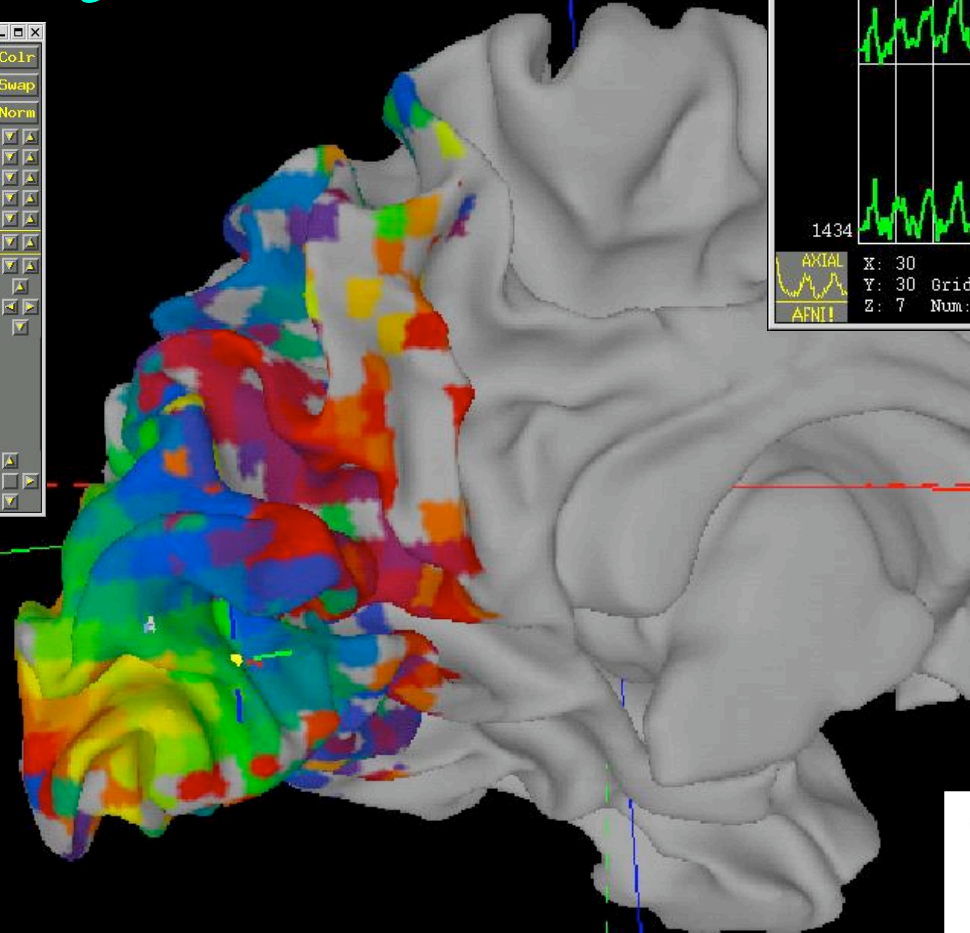
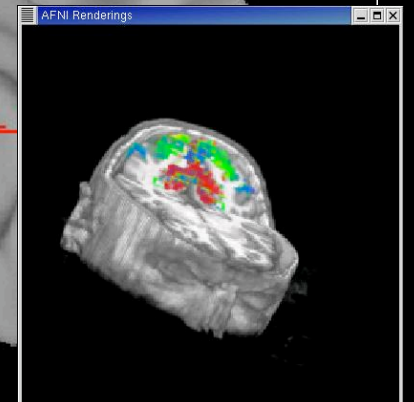
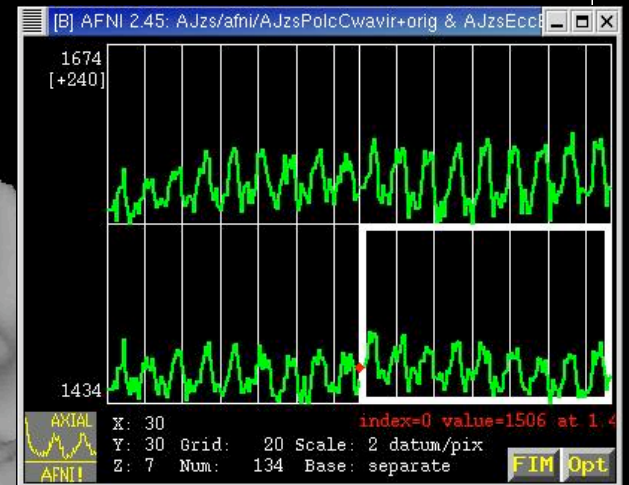
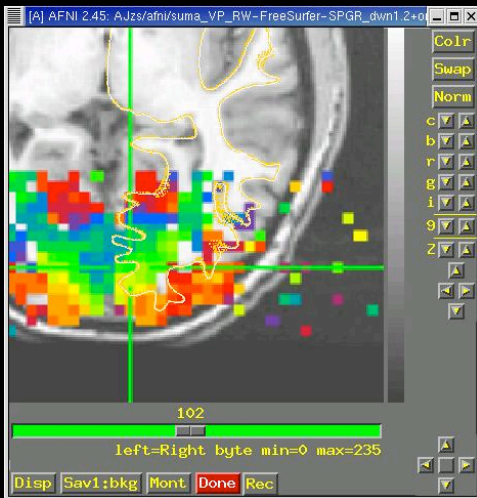
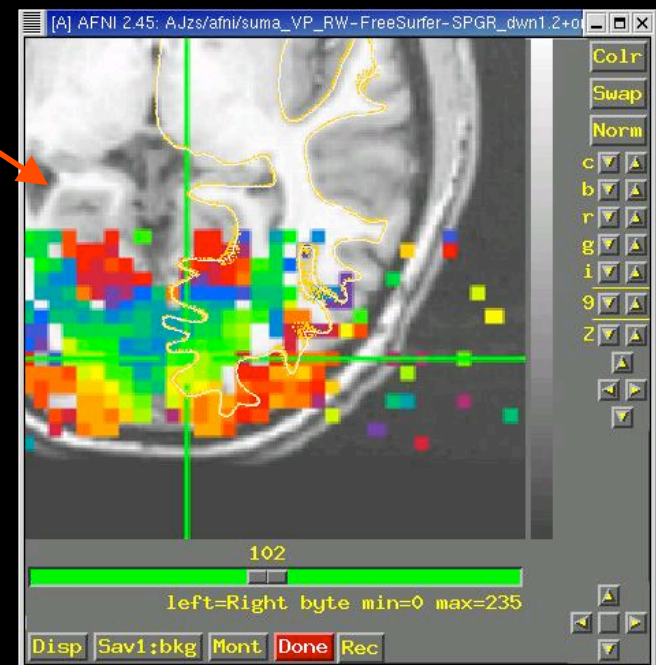


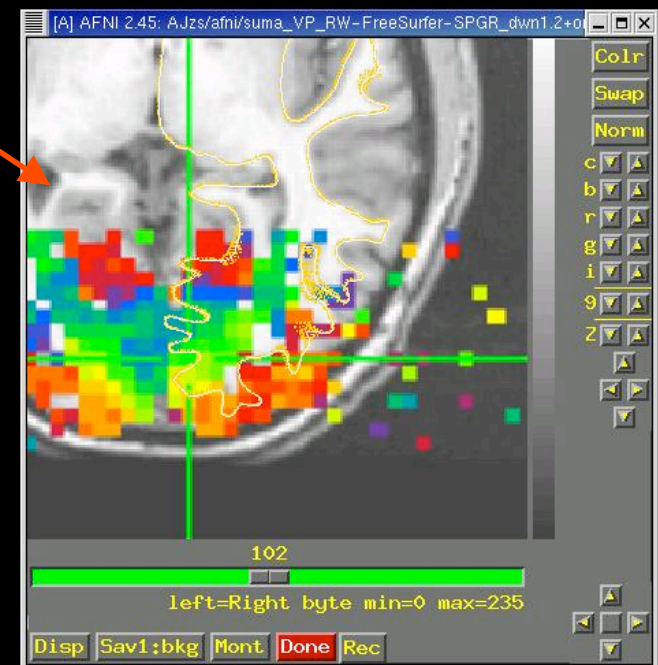
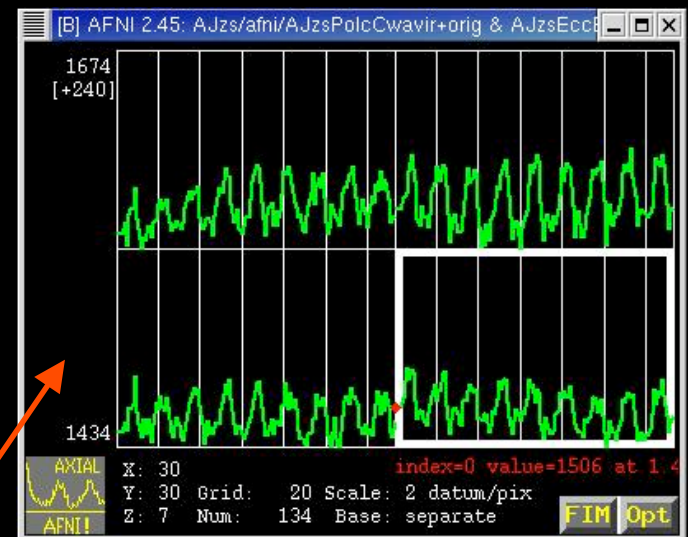
SUMA

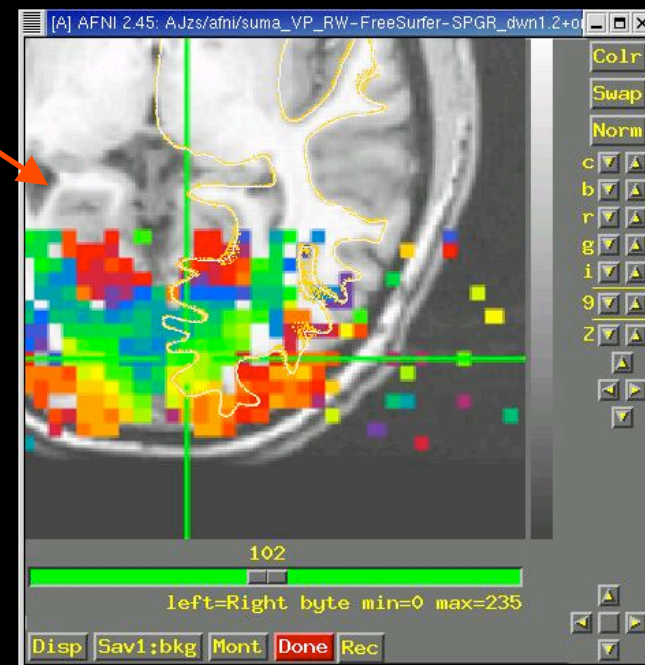
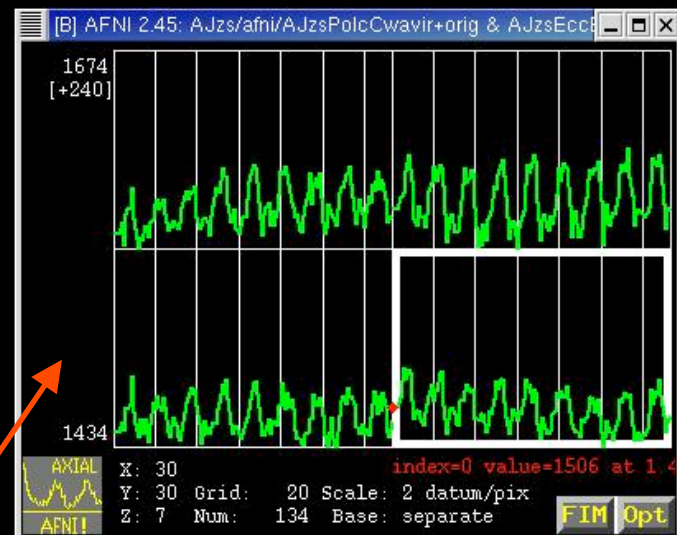
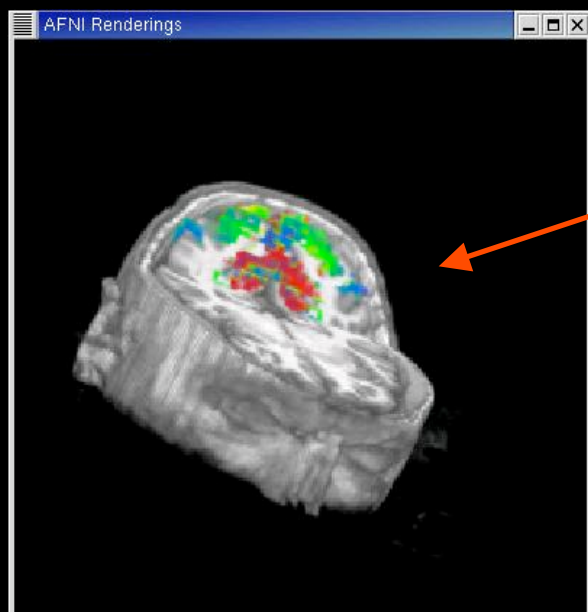


Statistical & Scientific Computing Core

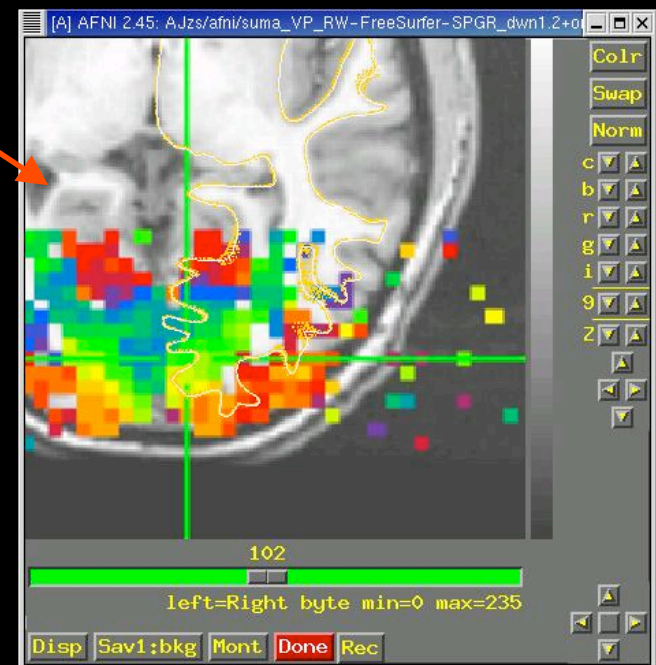
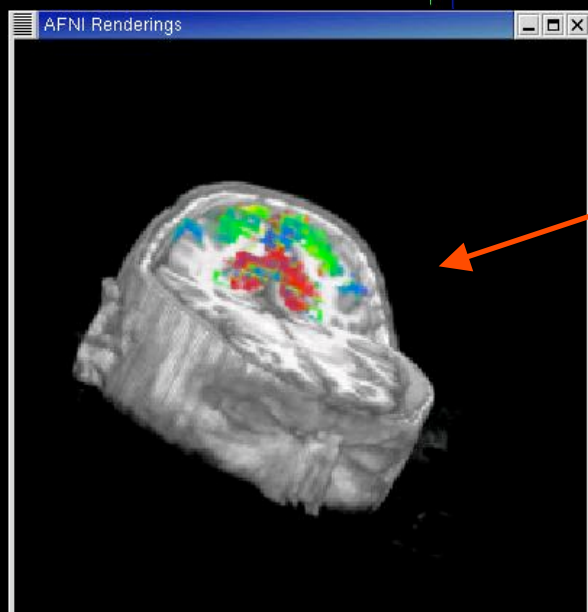
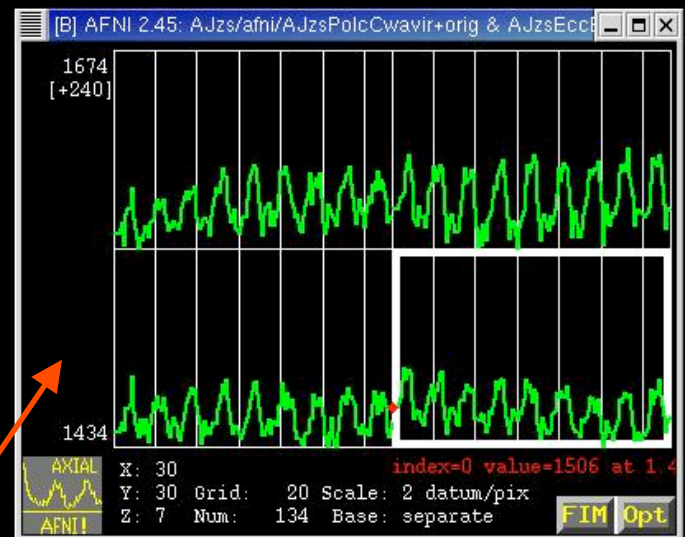
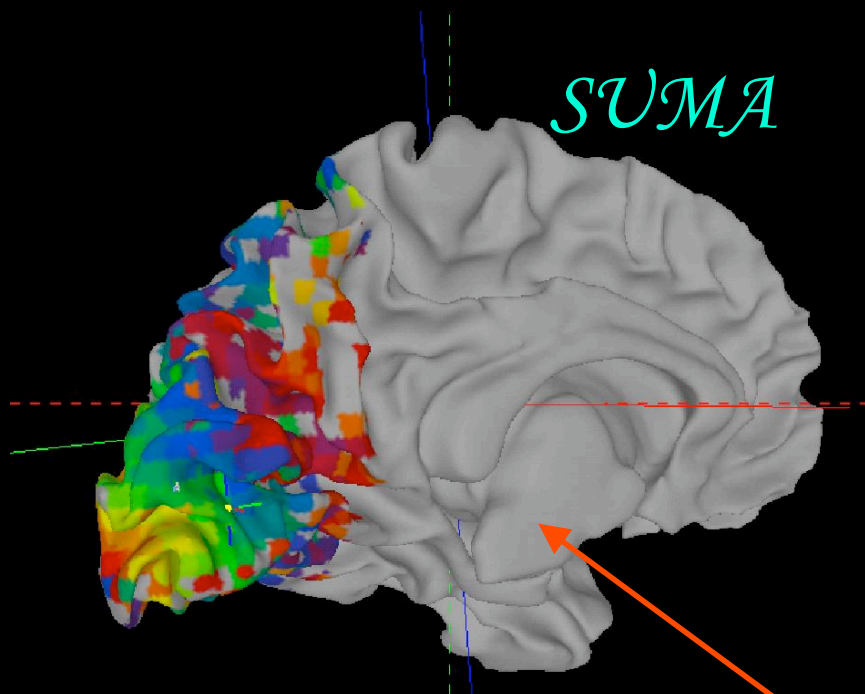








SUMA



SURface MApping with AFNI

- Surface mapping & viewing program tightly linked to AFNI
- Complements AFNI's slice and volume rendering modes
- Provides a framework for fast and user-customizable surface based analysis
- Supports surface models created by:
FreeSurfer <http://surfer.nmr.mgh.harvard.edu>
SureFit/Caret <http://stp.wustl.edu/resources/display.html>
BrainVoyager <http://www.brainvoyager.com>
- Allows representation of 2D and 3D objects in projection frame or surface coordinate space.

The Process

•PreSUMA:

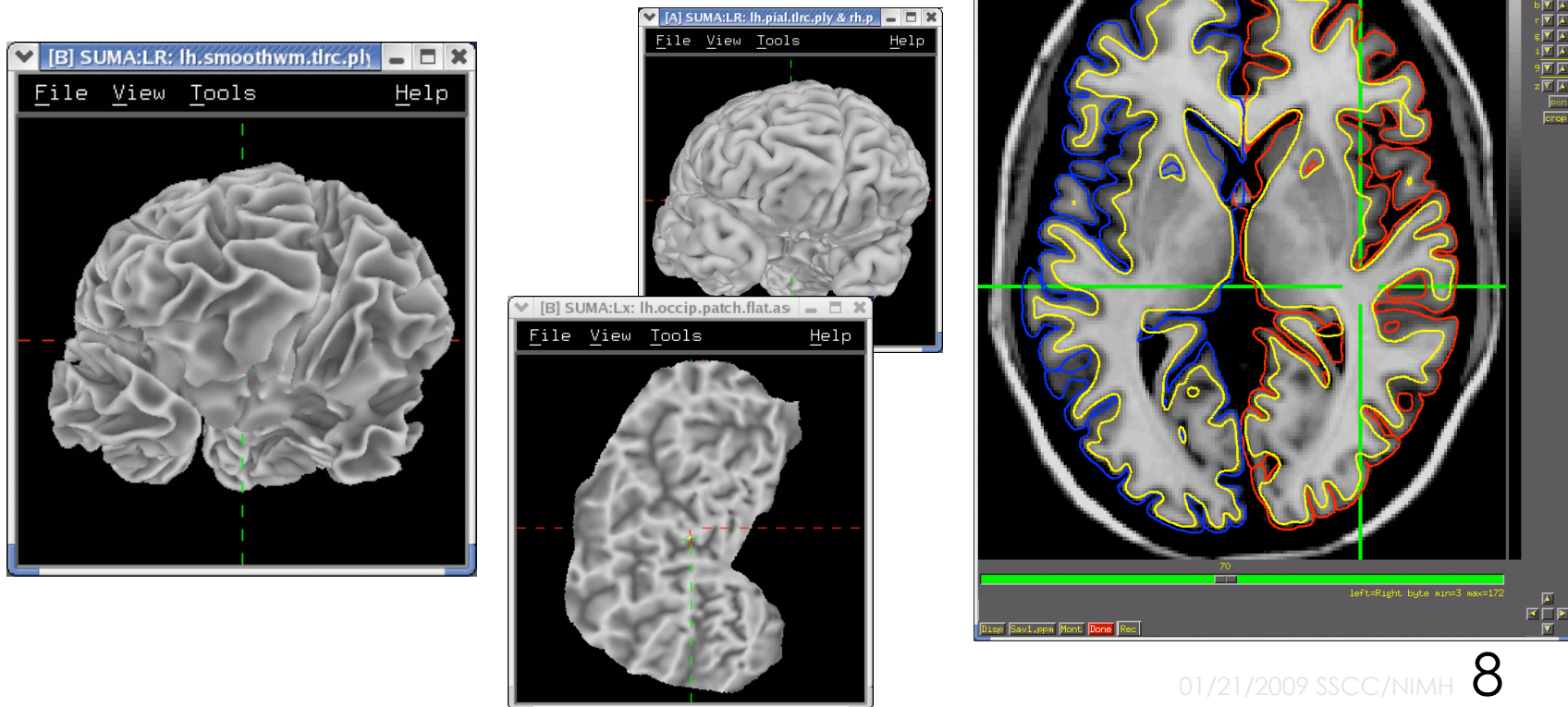
- Collect, align and average high-quality, high-resolution anatomical data
 - On NIMH's 3T-1, 4 MPRAGE data sets will do
- Correct image non-uniformity
 - Using AFNI's 3dUniformize or the N3 normalization tool [J.G. Sled et al. 98]
- Create and correct surfaces
 - Using FreeSurfer, SureFit or BrainVoyager

•CircumSUMA:

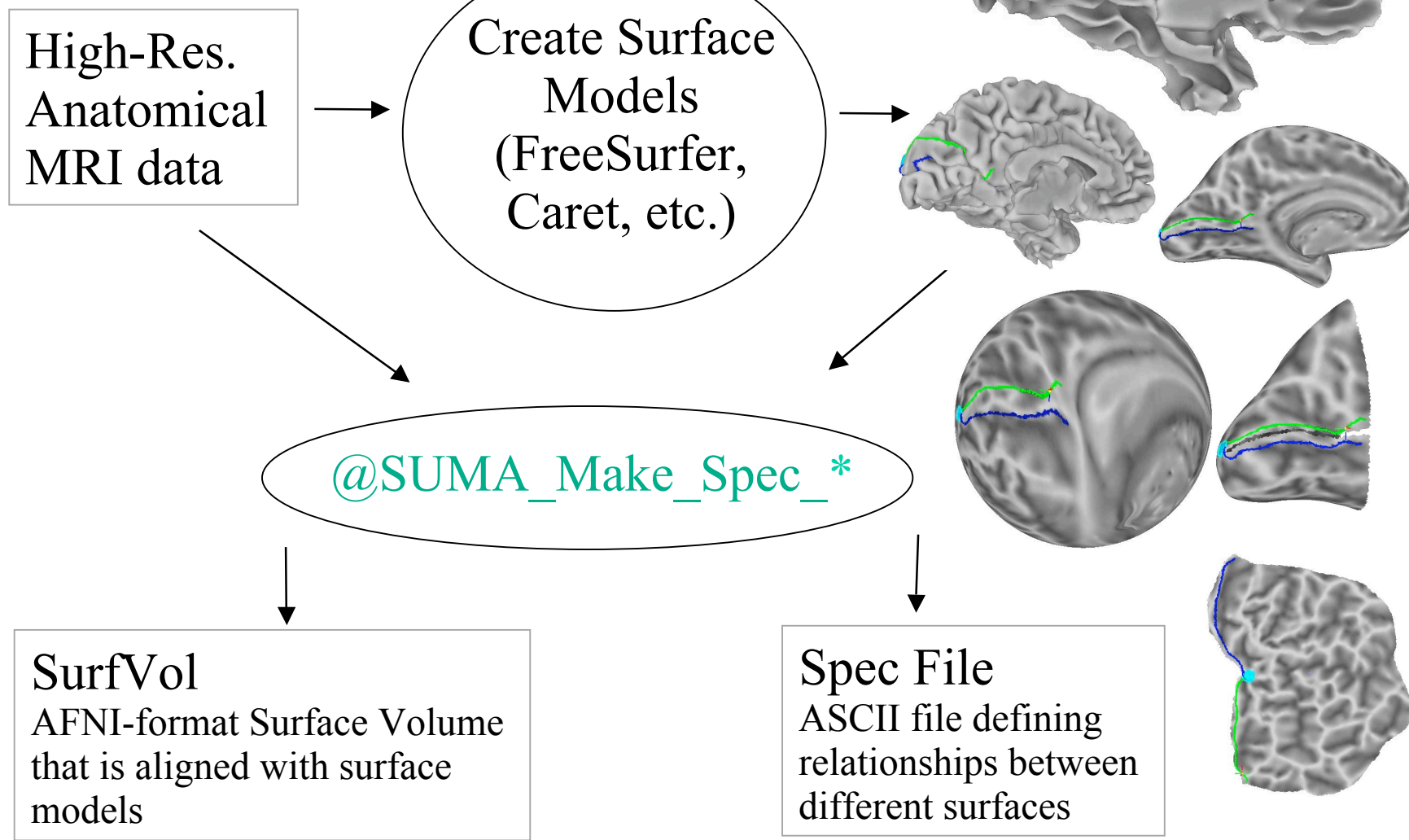
- Create standard-mesh version of surface models
- Align surface with experimental data
 - Using @SUMA_AlignToExperiment
- Map experimental volumetric data to surface
 - Using AFNI and SUMA
- Time series analysis
 - Smoothing, statistics, clustering, group analysis.

The Lesser Process

- For display (mostly) of Talairach data:
 - Use the Talairach surfaces created from the N27 brain data set using FreeSurfer.
 - Ready to use, no surface creation or alignment needed.



A: Preparing surface models for SUMA



@SUMA_Make_Spec_* scripts

- These scripts prepare surfaces and volumes for use with SUMA:
 - ★ @SUMA_Make_Spec_FS for FreeSurfer surfaces
 - ★ @SUMA_Make_Spec_SF for SureFit surfaces
 - ★ @SUMA_Make_Spec_Caret for Caret surfaces
- Output of these scripts include:
 - ★ Surface Volume: SurfVol
 - ★ An AFNI data set created from anatomical MRI used to create the surfaces
 - ★ AFNI volumes of cortical and subcortical segmentations (FreeSurfer)
 - ★ ASCII versions of all surfaces (FreeSurfer)
 - ★ Surface specifications file: (*.spec)
 - ★ An ASCII file defining relationships between family of surfaces.

Viewing Surfaces with SUMA

- After `@SUMA_Make_Spec_*` scripts are run, surface models should be in excellent alignment with SurfVol
 - ★ Use SUMA to verify alignment
 - ★ Scroll through the volume to make sure surfaces are accurate
 - ↳ Check especially for inferior temporal and occipital areas
- Demo using FreeSurfer surfaces:
 - ★ `cd suma_demo/SurfData/SUMA`
 - ↳ This is where the output of `@SUMA_Make_Spec_FS` resides
 - ★ `afni -niml &`
 - ↳ launches AFNI to allow the viewing of `DemoSubj_lh.spec`
 - ↳ the `-niml` option tells AFNI to listen to connections from SUMA
 - ★ `suma -spec DemoSubj_lh.spec -sv DemoSubj_SurfVol+orig &`
 - ↳ or execute the script: `tcsh run_suma`

Check for proper alignment and defects

- With both SUMA and AFNI running
 - ★ Press 't' in the suma window to 'talk' to AFNI
 - ★ This sends anatomically correct surface to AFNI
 - ★ **Switch Underlay** to DemoSubj_SurfVol+orig
 - ★ Contours are the intersection of the surface with the volume.
 - ⇒ You could also see boxes representing the nodes that are within +/- 1/2slice from the center of the slice in view.
 - ⇒ Colors and node box visibility can be changed to suit your fancy from the **Control Surface** button in AFNI.
 - ★ Navigate through the volume in AFNI.
 - ➡ make sure you have an excellent alignment between volume and surface
 - ➡ make sure surface adequately represents areas of the brain that are difficult to segment
 - ⇒ occipital cortex
 - ⇒ inferior frontal and inferior temporal regions

Check for proper alignment and defects

- The Surface Volume and the surfaces must be in perfect alignment.
 - ★ If you have an improper alignment, it should be addressed here.
 - ➡ This should not happen for FreeSurfer and SureFit surfaces created in the standard fashion.
 - ★ Watch for error messages and warnings that come up in the shell as the surfaces are read in. These messages should be screened once since they do not change unless the surface's geometry or topology is changed.
 - ★ Viewed without the volume underlay, it is extremely difficult to tell if surface models with no topological defects accurately represent the cortical surface.

Basic SUMA viewer functions

- Rotating the surface:
 - ★ **Mouse button-1:** keep it down while moving the mouse left to right. This rotates the surface about the screen's Y-axis (dotted green). Let go of button-1.
 - ★ Repeat with up and down motion for rotation about X-axis and motion in various directions for rotations mimicking those of a trackball interface.
 - ★ Also try up/down/left/right arrow keys.
 - ➔ Arrow keys rotate by increments specified by the environment variable: SUMA_ArrowRotAngle (degrees).

Basic SUMA viewer functions

- Translating the surface:
 - ★ **Mouse button-2:** keep it down while moving the mouse to translate surface along screen X and Y axes or any combinations of the two.
 - ★ Also try shift+arrow keys.
- Zooming in/out:
 - ★ **Both buttons 1&2 or Shift + button 2:** while pressing buttons, move mouse down or up to zoom in and out, respectively.
 - ★ Also try keyboard buttons 'Z' and 'z' for zooming in and out, respectively.

Basic SUMA viewer functions

- Picking a Node or Facet:
 - ★ **Mouse button 3:** press over a location on a surface to pick the closest facet and node to the location of the pointer.
 - ➔ The closest node is highlighted with a blue sphere
 - ➔ The closest facet is highlighted with a gray triangle
 - ★ Note the information written to the shell regarding the properties of the picked Node and Facet.
 - ★ When connected to AFNI (after having pressed 't'), watch the AFNI crosshair jump to the corresponding location in the volume.
 - ★ Conversely, position the crosshair in AFNI (left click) at a position close to the surface and watch the crosshair relocate in SUMA.
- You can swap button 1 & 3's functions using the environment variable: SUMA_SwapButtons_1_3

Basic SUMA viewer functions

- Cardinal views:
 - ★ ctrl + Left/Right: Views along LR axis
 - ★ ctrl + Up/Down: Views along SI axis
 - ★ ctrl + shift + Up/down: Views along AP axis
- Resetting the view point:
 - ★ Press Home to get back to the original vantage point.
- Using momentum feature:
 - ★ Press 'm' to toggle momentum on. Click the left mouse button and release the button as you are dragging the mouse.
- Lots more:
 - ★ Function keys modify various aspects of the display
 - ★ ctrl+h opens a help window for all interactive options.

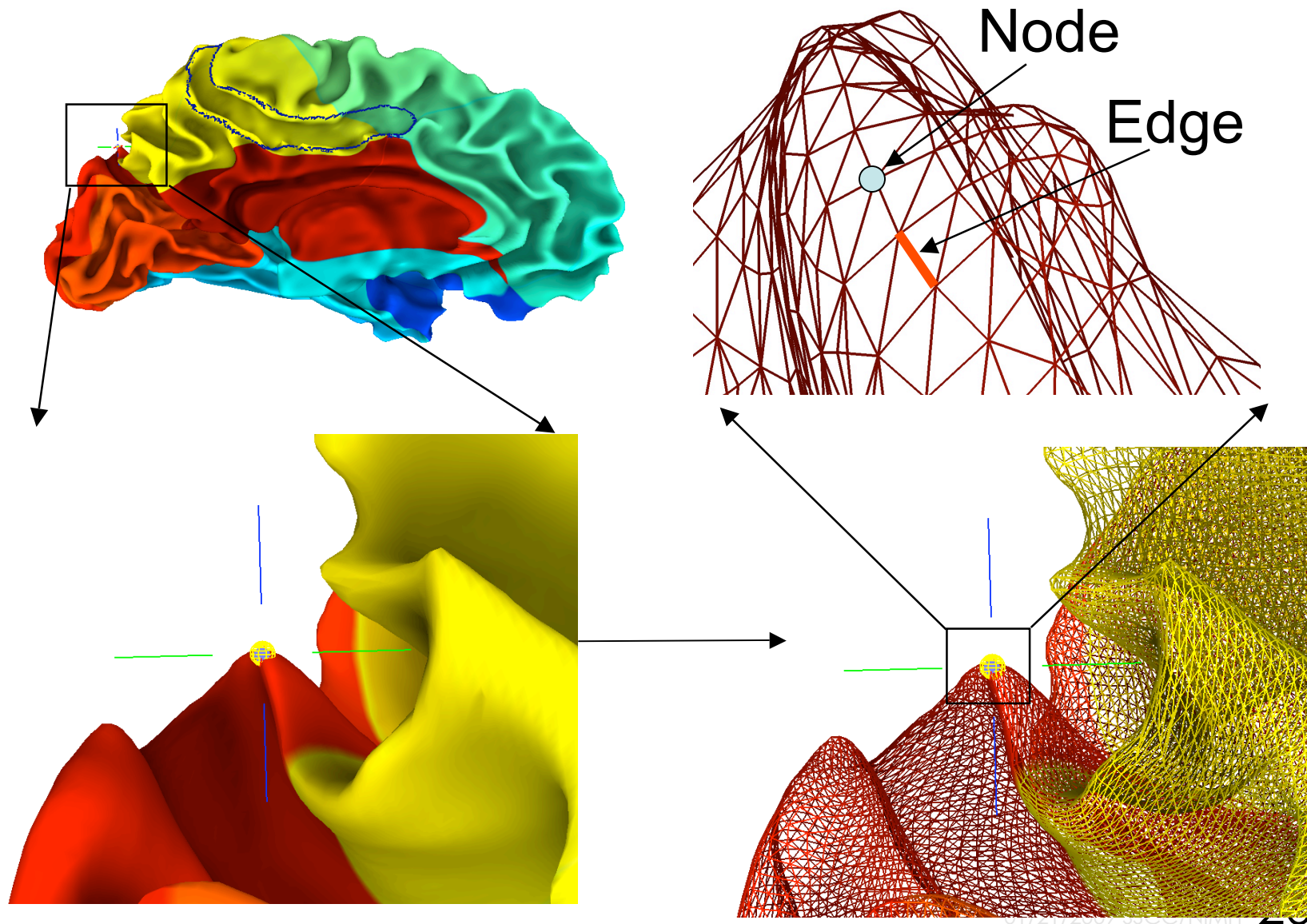
Recording your rendered images

- Using 'r' in SUMA to record the current scene.
- Using 'r' on the colorbar creates an image of the colorbar.
- Using 'R' to record continuously the rendered scene.
- Images are captured by an AFNlesque viewer.
 - ★ Identical consecutive images are rejected
 - ★ Images caused by X expose events are ignored
 - ★ Images can be saved in all ways allowed by AFNI, including MPEG and animated GIF
 - ★ If you let the recorder run continuously with very large images, you might quickly run out of memory
- You can save/load viewer setting used to create figure
 - ★ Use 'File→Save' View and 'File→Load View'

world \leftrightarrow AFNI \leftrightarrow SUMA \leftrightarrow world

- AFNI and SUMA are independent programs and communicate using NIML formatted data elements
 - ★ Via shared memory or network sockets
 - ★ Both AFNI and SUMA can also communicate with other programs
- NIML: NeuroImaging Markup Language developed by Dr. R.W. Cox
 - ★ NIML is the main format for SUMA's data storage
- NIML API library for packing/unpacking data is available
- Protocol allows independent program to communicate with AFNI
- Advantages include:
 - ★ Programs execute on separate machines
 - ★ Fast development
 - ★ Screen real-estate
- Blemishes include:
 - ★ Only one AFNI can be listening for connections
 - ★ Only one SUMA can connect to AFNI

What's a surface made of?

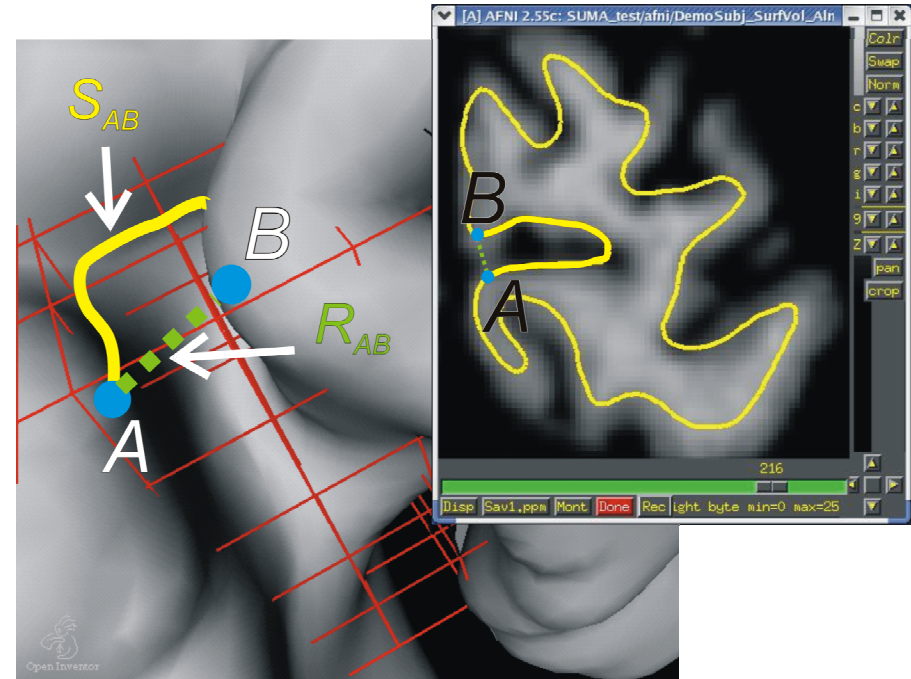


Relationships between surface models

- Surface Geometry:
 - ★ refers to the spatial coordinates of nodes forming a surface model
- Surface Topology:
 - ★ refers to the connectivity between nodes forming a surface model
- A family of surfaces with different geometry but similar topology is created for each surface model.
 - ★ white/grey, pial, inflated, spherical, flattened, etc.
- Some models' geometries are anatomically correct
 - ★ Pial and/or white matter surfaces can be used for relating to volume data.
 - ★ Inflated, flattened and spherical cannot be directly linked to volume data. The link is done via their corresponding anatomically-correct surfaces.
- Those relationships are encoded in the Spec file.

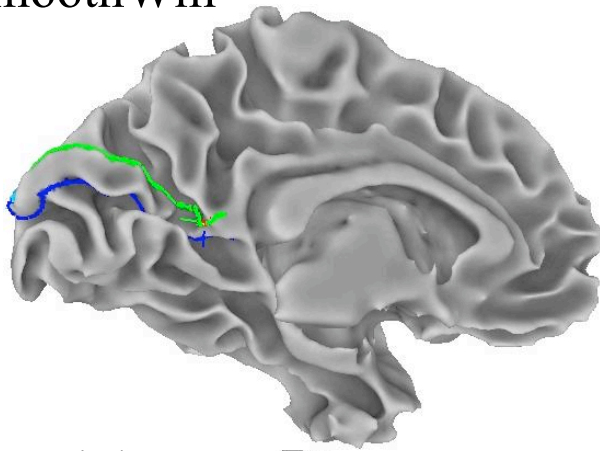
Geometry and Topology

- Geometry: Spatial location
 - ★ X,Y,Z coordinates of brain structures
- Topology: Spatial connectivity
 - ★ Relative positions of brain structures along the surface

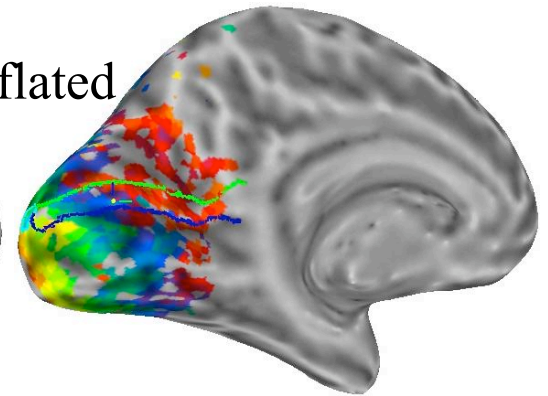
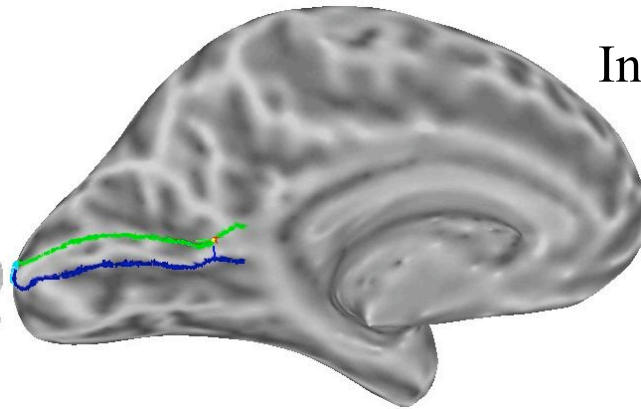


- Geometric proximity does not imply topological proximity
 - If you care about the topology of activation, you should transfer FMRI data onto the surface before spatial manipulations of the data

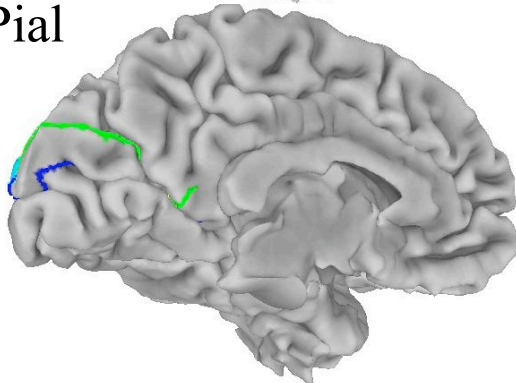
SmoothWm



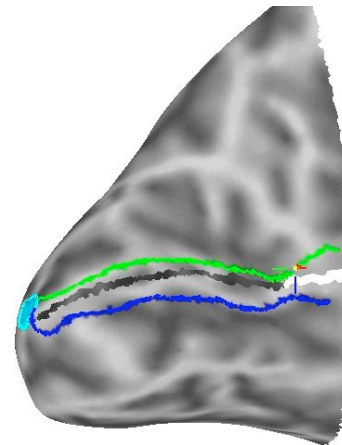
Inflated



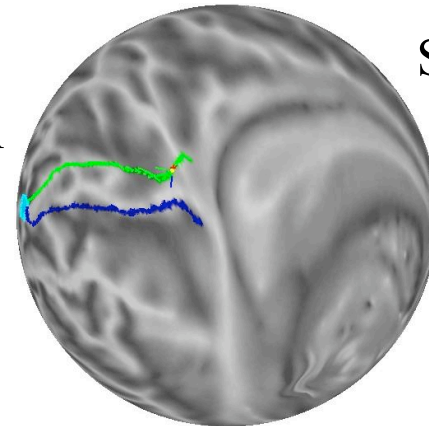
Pial



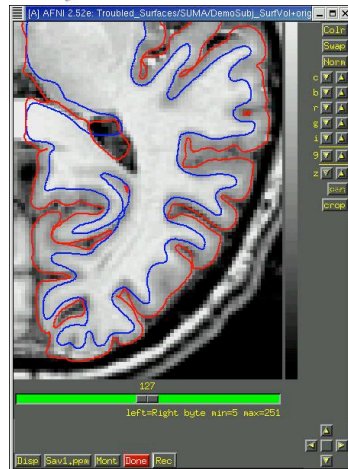
Inflated,
Occipital
cut



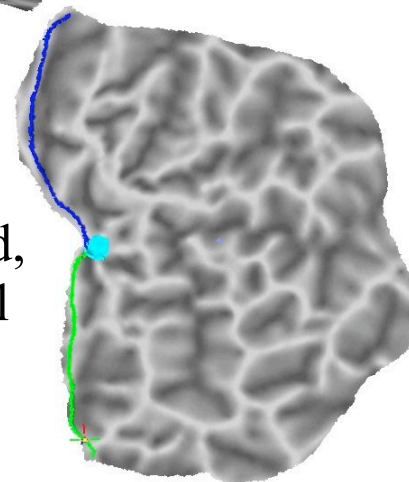
Spherical



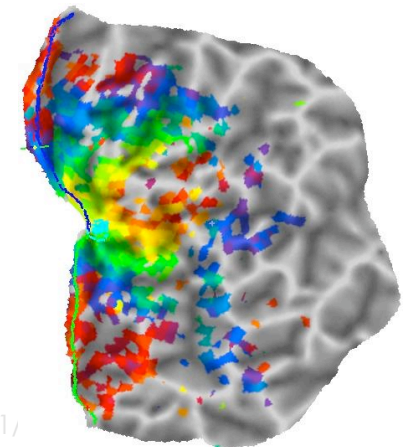
Overlay of
anatomically
correct Pial and
SmoothWm
surfaces over
anatomical
volume



Flattened,
Occipital
cut



21/



Viewing the group of surfaces

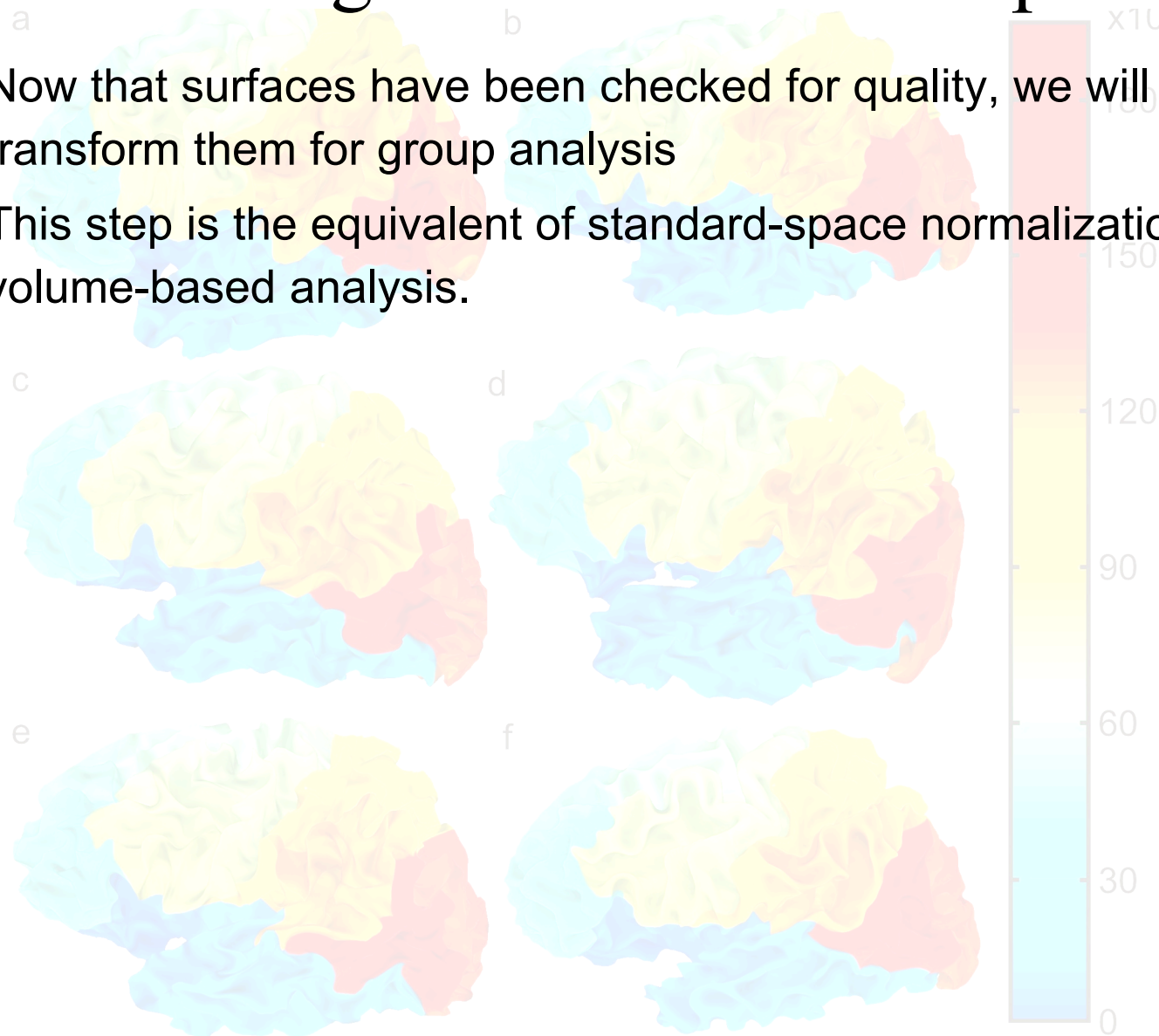
- Switch Viewing States:
 - ★ '.' switches to the next viewing state (pial then inflated etc.)
 - ★ ',' switches to the previous viewing state
 - ★ Navigate on any of the surfaces and watch AFNI's crosshair track surface
 - ★ SPACE toggles between current state and Mapping Reference state
- Viewing multiple states concurrently:
 - ★ 'ctrl+n' opens a new SUMA controller (up to 6 allowed, more possible but ridiculous)
 - ★ switch states in any of the viewers
 - ★ all viewers are still connected to AFNI

Viewing the group of surfaces

- Controlling link between viewers:
 - ★ Open SUMA controller with 'ctrl+u' or View->SUMA Controller
 - ★ SUMA controller crosshair locking options:
 - ↳ '-': no locking
 - ↳ 'i': node index locking (i.e. topology based)
 - ↳ 'c': node coordinate locking (i.e. geometry based)
 - ★ SUMA controller view point locking
 - ↳ 'v': depress toggle button to link view point across viewers.
 - ⇒ Surface rotation and translation in one viewer is reflected in all linked viewers

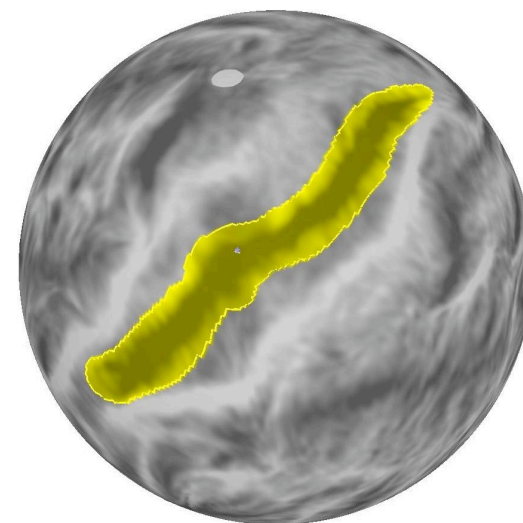
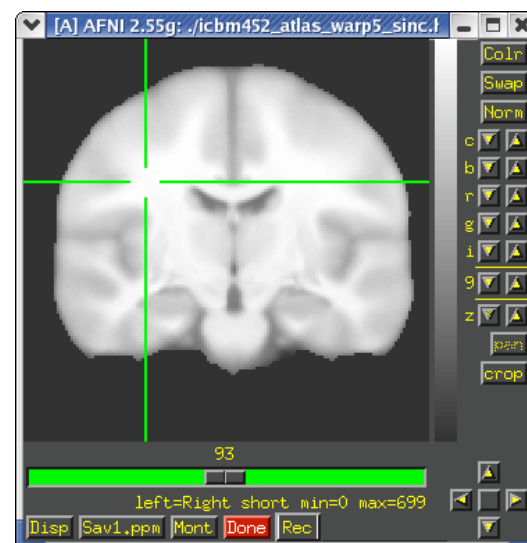
Standardizing Surfaces For Group Analysis

- Now that surfaces have been checked for quality, we will transform them for group analysis
- This step is the equivalent of standard-space normalization for volume-based analysis.

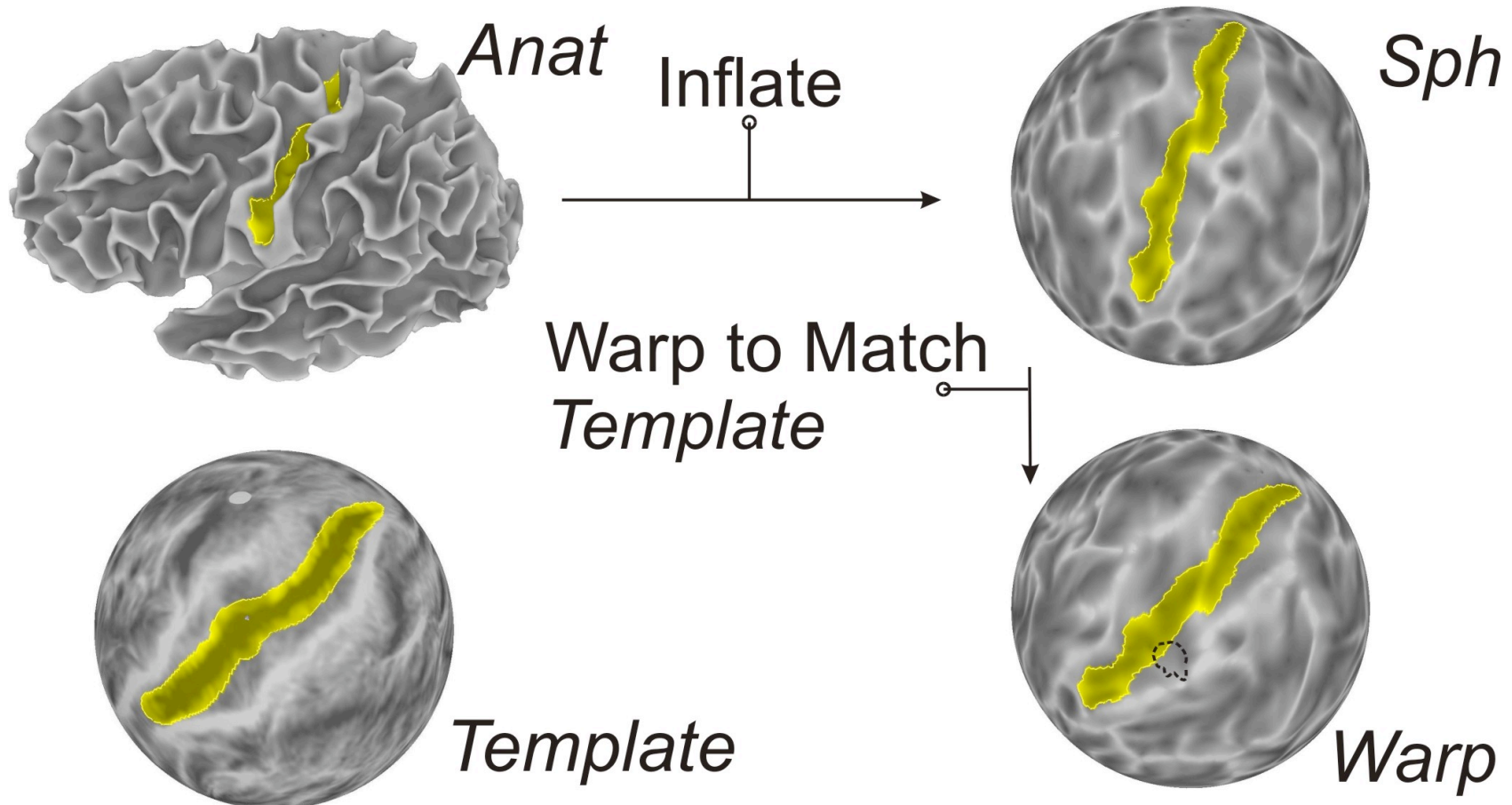


Group analysis

- Group analysis requires data defined over a domain common to all subjects
 - ★ Talairach space for volume-based analysis
 - ★ With low order registration
 - + Small amount of geometric distortion
 - No respect for topology of activation
 - ★ With high order registration
 - + Preserves the topology of action
 - Considerable geometric distortion
 - ★ Spherical coordinate system for surface-based analysis
 - + Preserves the topology of activation
 - Considerable geometric distortion
- Preserving the topology of activation is Good in general and crucial for
 - ★ Retinotopy
 - ★ Plasticity
 - ★ High-resolution Mapping



Spherical Warping



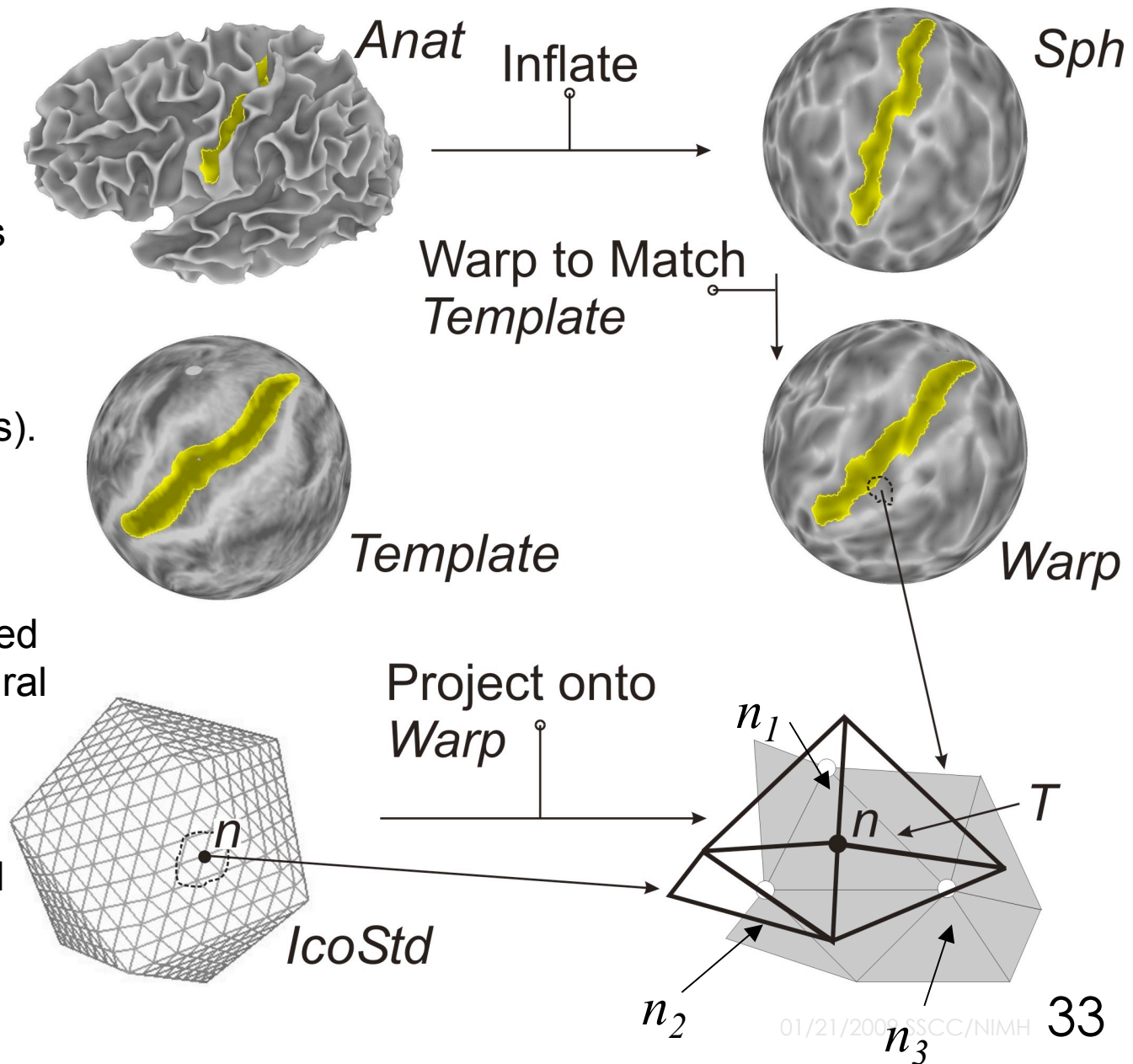
Surface-based warping is more accurate than the more common low order Talairach volume-based analysis because it preserves the topology of the cortical sheet and uses more landmarks for the warping

Mapping data

Problem is that surfaces from different subjects are not topologically isomorphic (different meshes).

Data from each subject are mapped onto the icosahedral surface for group analysis

Cumbersome and unnecessary interpolation

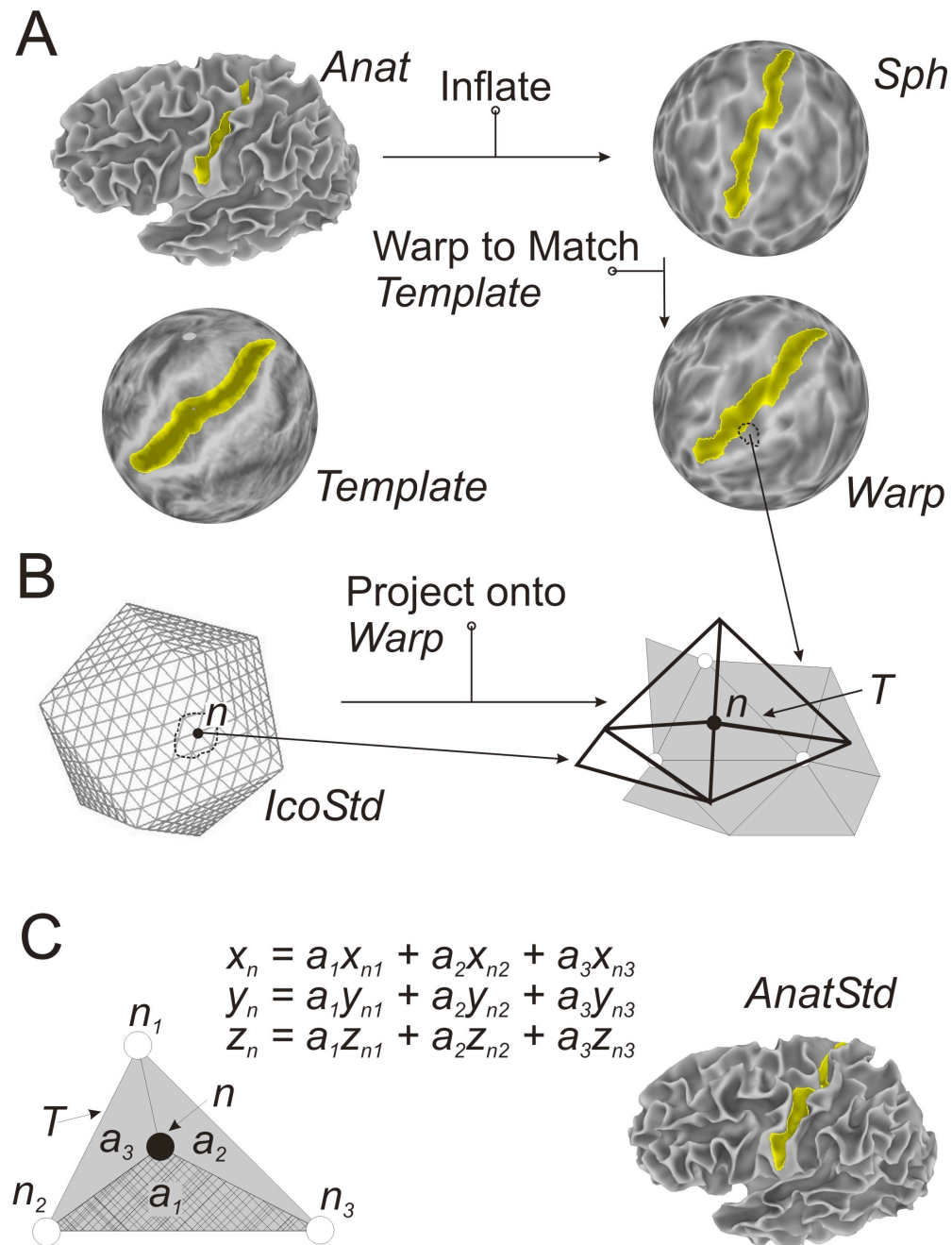


Our twist

Instead of interpolating data values to the icosahedral nodes, interpolate using the coordinates of the original surface's node.

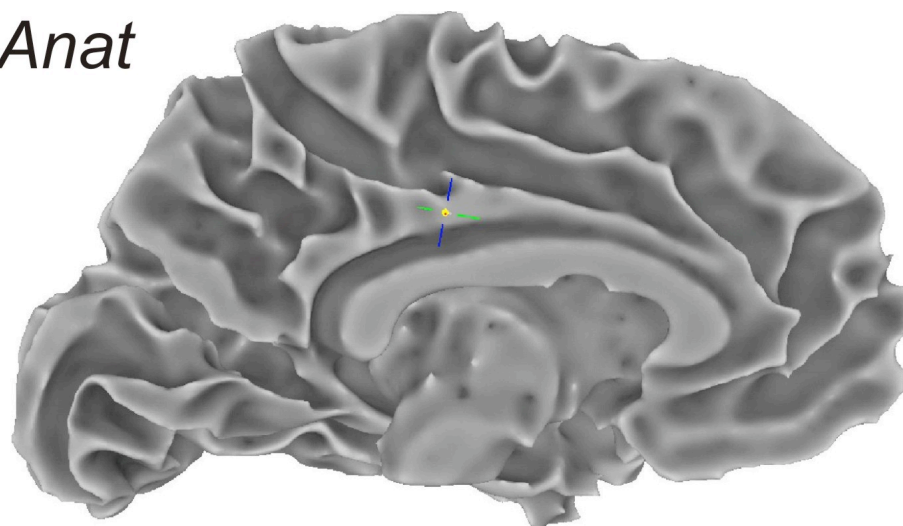
This results in a surface that is virtually identical in geometry to the original surface but with the mesh of the icosahedron.

Cross-subject surface based analysis is thus reduced to node-based analysis

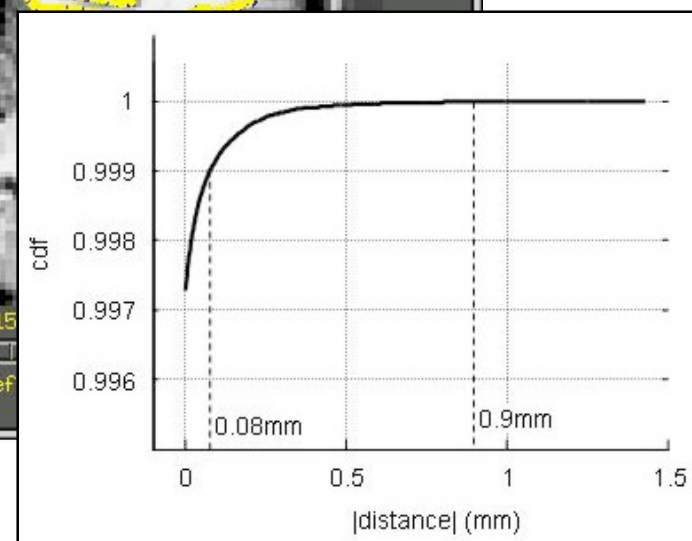
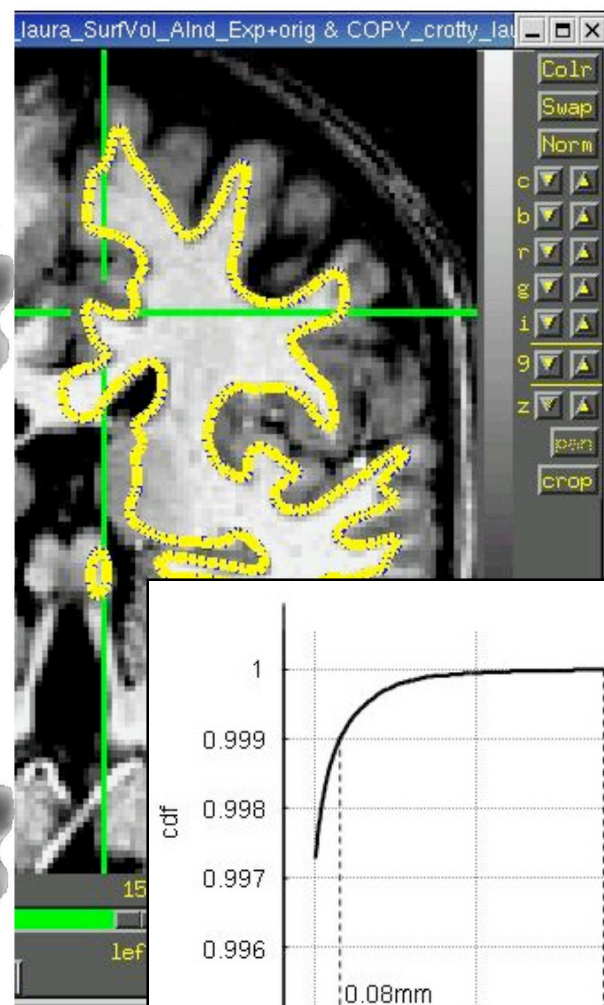
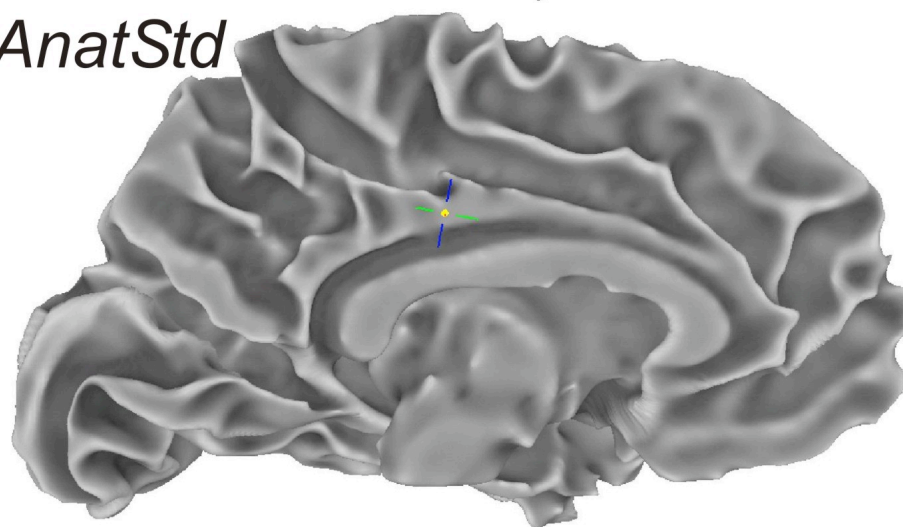


Compare surfaces

Anat



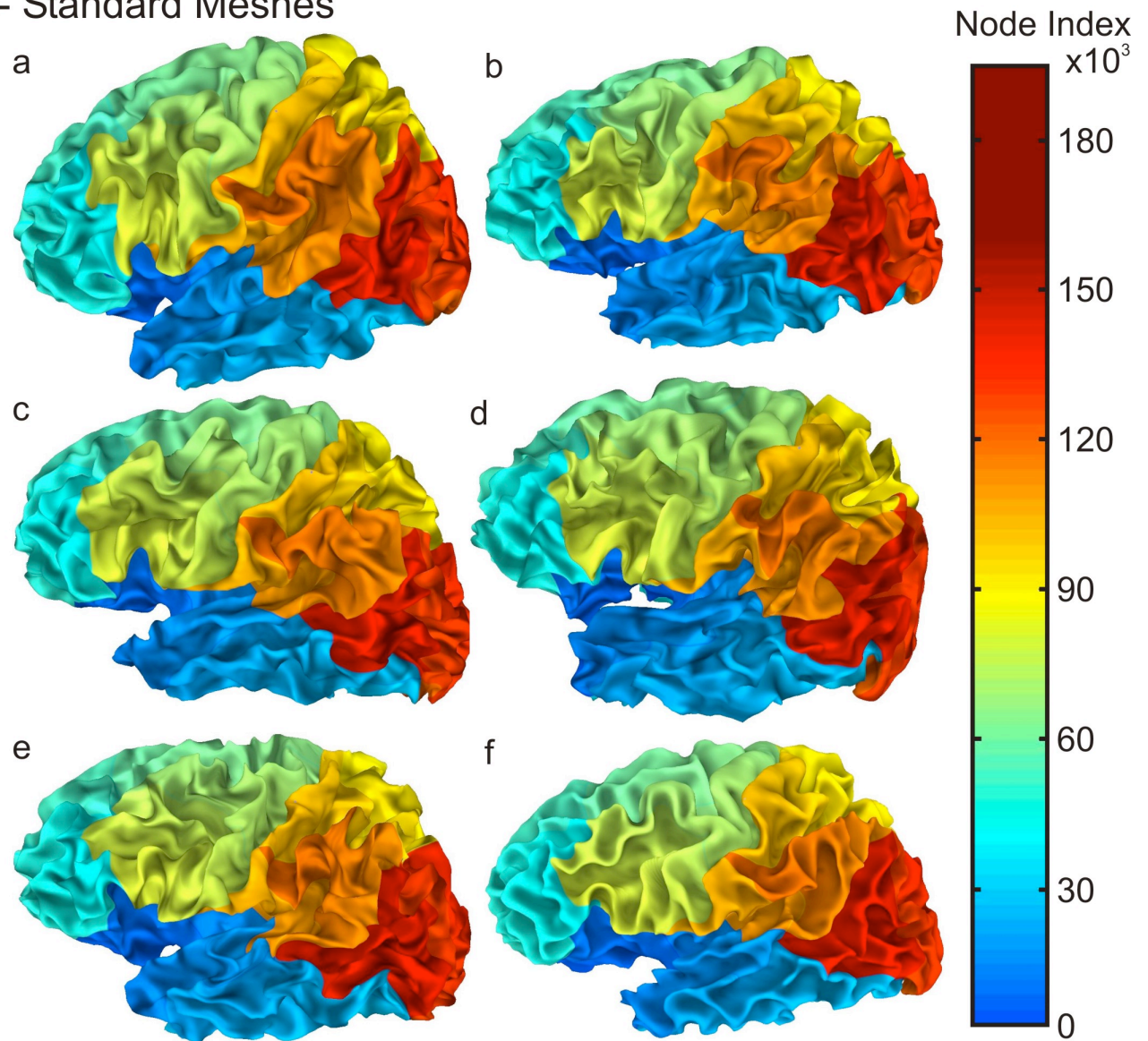
AnatStd



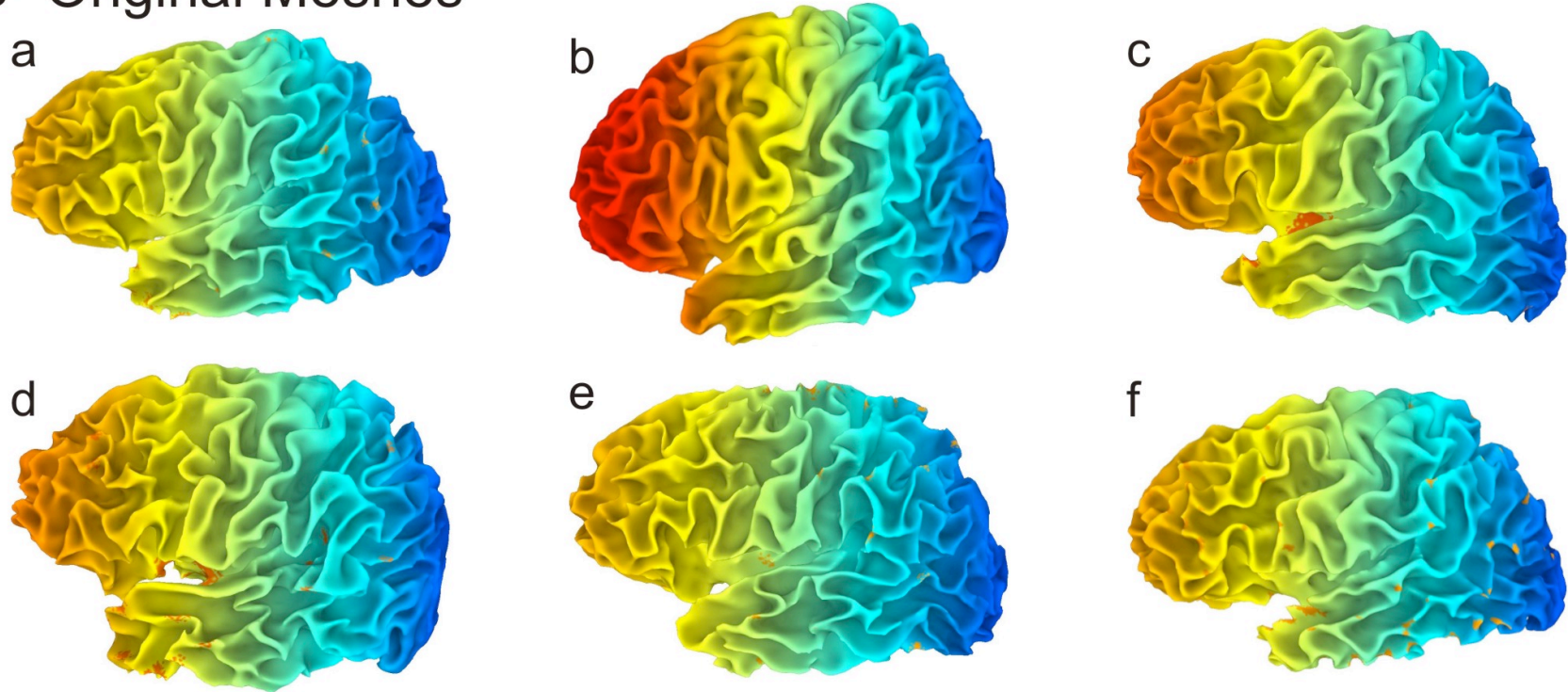
Standard meshes for 6 subjects

A- Standard Meshes

- 6 standard-mesh surface models from different subjects.
- Node colors encode for node index n on the standard mesh.
- Nodes with similar indices correspond to comparable sulcal landmarks despite anatomical variability across subjects.



B- Original Meshes



- Original-mesh versions of standard-mesh surfaces with same coloration scheme
- Surfaces have differing numbers of nodes, some colors may not be represented
- Nodes with similar indices no longer correspond to comparable anatomical areas
- Small clusters of red color show new nodes later added to correct topological errors

Creating Standard-Mesh Surfaces

- Required Data:
 - Original Surface models
 - Warped Spherical surface
 - Spec file of the surfaces above (i.e. DemoSubj_lh.spec)
- Creating the standard-mesh versions of the original surfaces


```
MapIcosahedron      -spec Demo_Subj_lh.spec    \    Written by B. Argall
                    -ld 141                      \
                    -prefix ld141
```

-spec SpecFile: option specifying spec file with original surfaces

-ld n : number of subdivisions for each edge of the icosahedron.

$N_v = 2 + 10n^2$	(198812 vertices)
$N_t = 20n^2$	(397620 triangles)
$N_e = 30n^2$	(596430 edges)

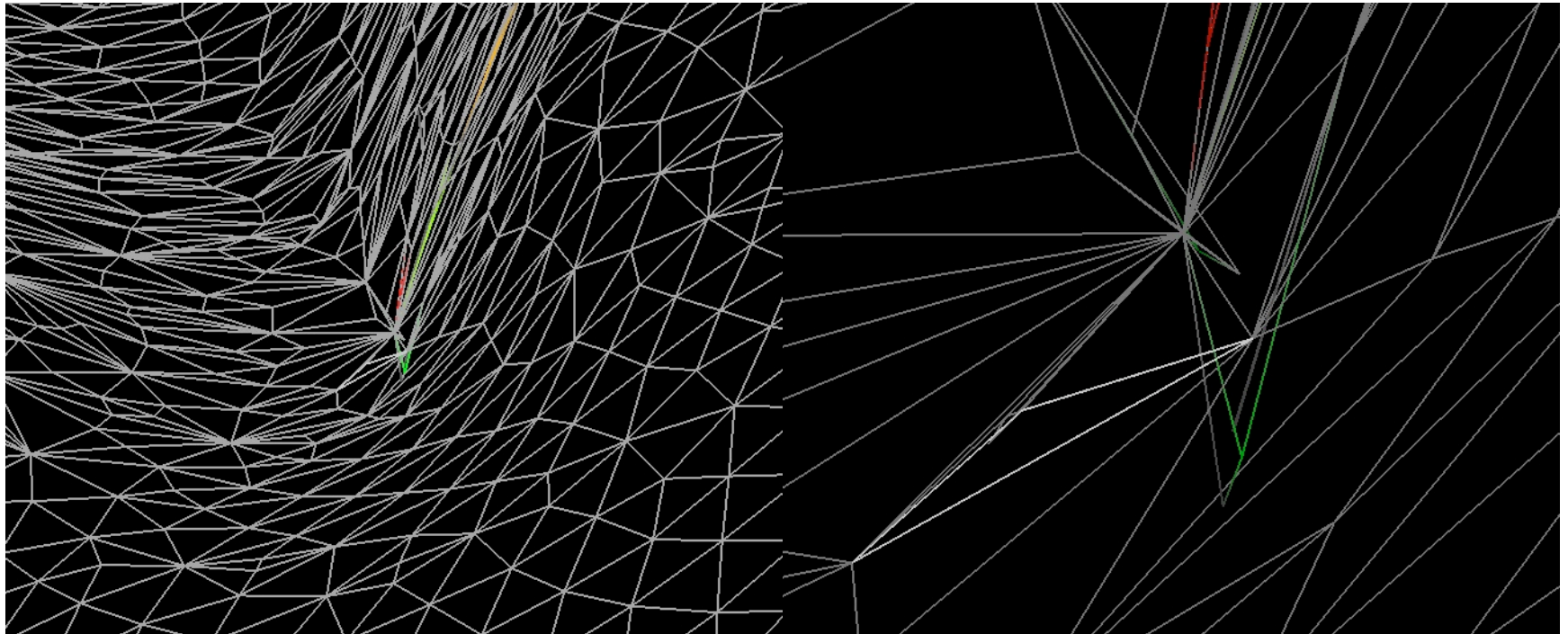
-prefix: prefix assigned to standard mesh surfaces
- Script *run_stdmeshes* was used to generate standard-mesh surfaces for the hands-on material

Using Standard-Mesh Surfaces

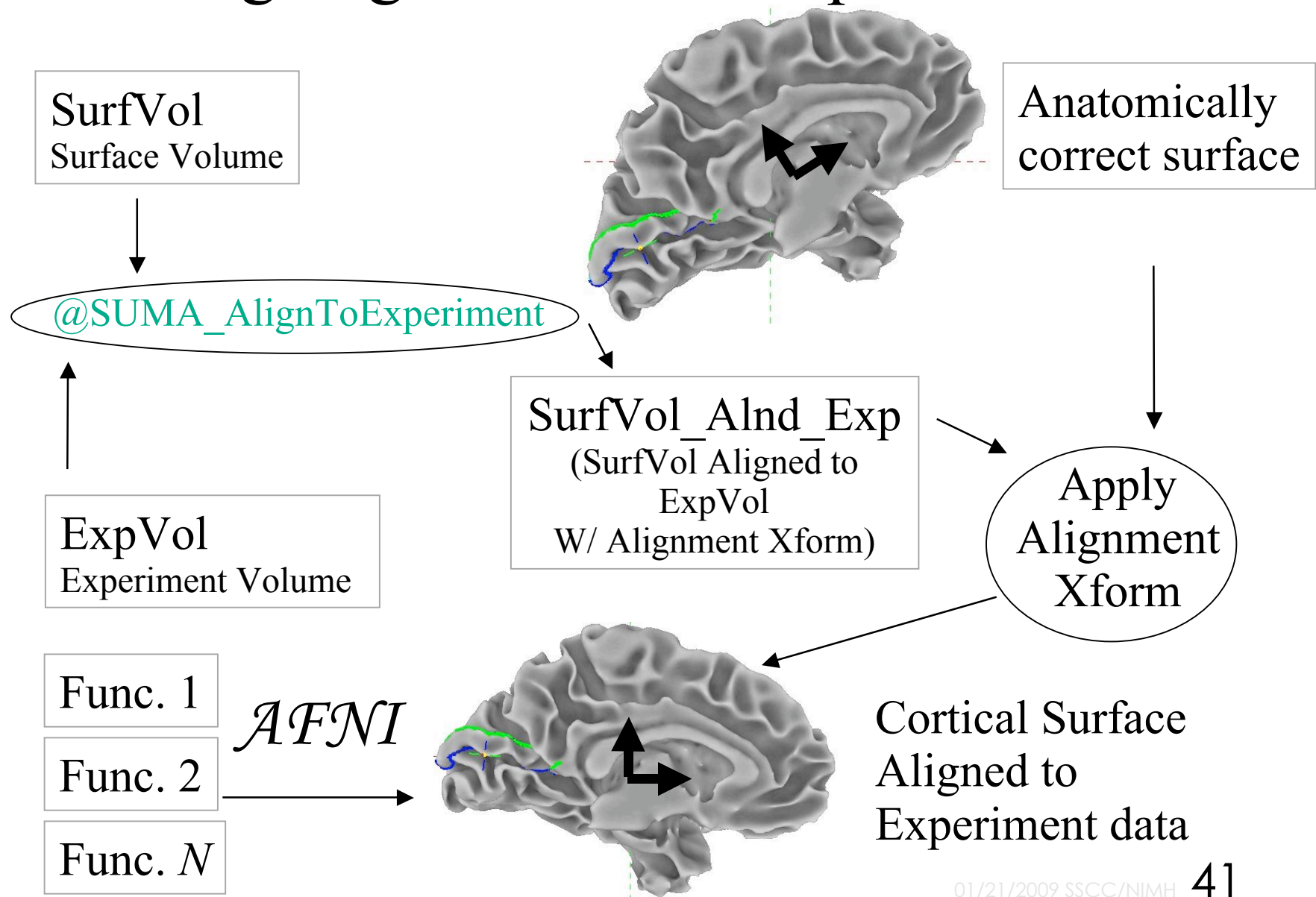
- Subsequent analysis will use standard-mesh surfaces, not originals
 - ★ Map single-subject data to corresponding standard-mesh surfaces.
 - ★ Use any of AFNI's voxel-based statistical tools to perform within- and cross-subject analysis in the surface domain
- Creating standard-mesh surfaces takes a couple of minutes per subject
- You can compare original and standard surfaces using:
 - SUMA
 - create a spec file containing both original and standard-mesh surfaces.
 - open two views, one showing original surface and one showing standard-mesh surfaces
 - link viewers by coordinates (not by node index) from the SUMA controller (ctrl+u)
 - CompareSurfaces Written by Shruti Japee
 - a program that calculates the distance from each node on surface 1 along its normal to surface 2
 - SurfaceMetrics
 - a program that calculates metrics of the mesh like edge lengths, triangle areas, curvature, etc.

Artifacts with standard meshes

- In the past, there have been localized distortions between the standard-mesh surface and the original surface.
- These had all been caused by topological errors in the original spherical surfaces.
- SurfQual was written to highlight these errors
 - ★ if causing distortions, errors must be fixed with the program used to generate them.
 - ★ These topological errors can be ignored if they do not cause any visible distortions in the standard-mesh surfaces.



B: Aligning Surface w/ Experiment Data



B: Aligning Surface w/ Experiment Data

- Functional data are assumed to be in register with experiment's anatomical
 - ★ Align Anatomical to EPI using: [align_epi_anat.py](#).
- Surface Volume is aligned to experiment's anatomical volume with rigid-body or affine transformation
 - ★ The script [@SUMA_AlignToExperiment](#) simplifies this step and rarely fails. If it does, you could try manual registration with [3dTagalign](#)
 - ★ Brain coverage and image types should be comparable, not necessarily identical
- Functional data are not interpolated

B: Aligning Surface w/ Experiment Data

- Demo: (close previous SUMA and AFNI sessions)

★ *cd suma_demo/afni*

- ➔ DemoSubj_spgrax+orig (experiment's high-res. anatomical scan)
- ➔ DemoSubj_EccExpavir+orig & DemoSubj_EccExpavir.DEL+orig (EPI timeseries and function.)

★ To perform the alignment we ran:

```
@SUMA_AlignToExperiment \
    -exp_anat          DemoSubj_spgrsa+orig \
    -surf_anat         ../SurfData/SUMA/DemoSubj_SurfVol+orig
```

- ➔ This script 3dvolreg or 3dAllineate to align the experiment's anatomical volume to the Surface Volume.
- ➔ The script takes care of resampling (with 3dresample) the experiment's anatomical volume to match the Surface Volume if need be.
- ➔ The output volume is named with the prefix of the Surface Volume with the suffix _Alnd_Exp (read Aligned to Experiment).

B: Aligning Surface w/ Experiment Data

- Launch AFNI to check that volumes aligned well:
 - *afni -niml &*
 - Switch Underlay to DemoSubj_SurfVol_Alnd_Exp+orig
 - Switch Overlay to DemoSubj_spgrsa+orig
 - Visually check the alignment:
 - For example, with pointer in slice viewing window:
 - press 'o' on the keyboard to turn off the overlay
 - press 'u' repeatedly to toggle between the two volumes

★ Launch SUMA to check alignment of surface with volumes

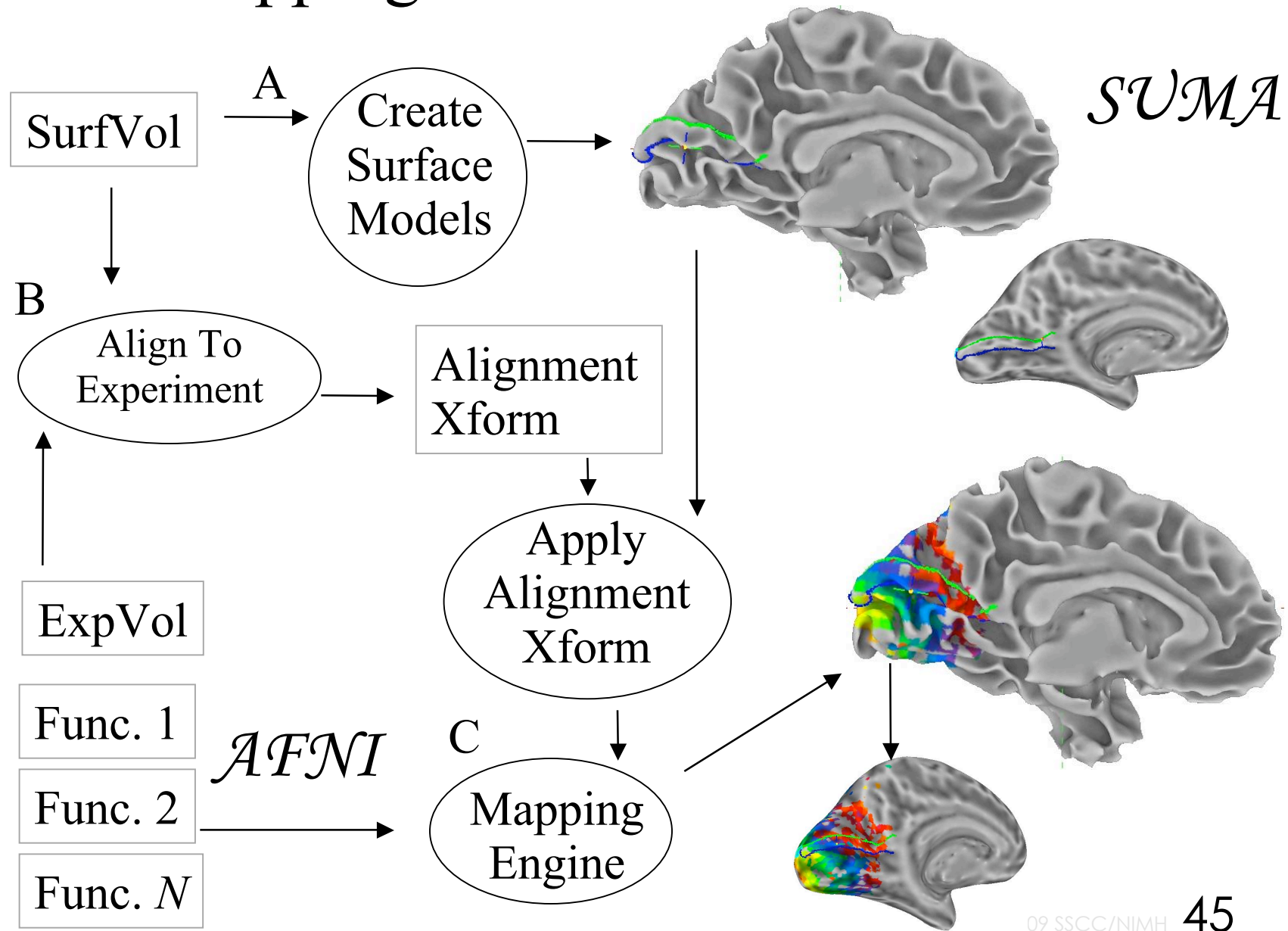
```
suma      -spec ../SurfData/SUMA/std.DemoSubj_lh.spec \  
          -sv DemoSubj_SurfVol_Alnd_Exp+orig &
```

➔ *or execute the script: tcsh run_suma*

Press 't' to talk to AFNI

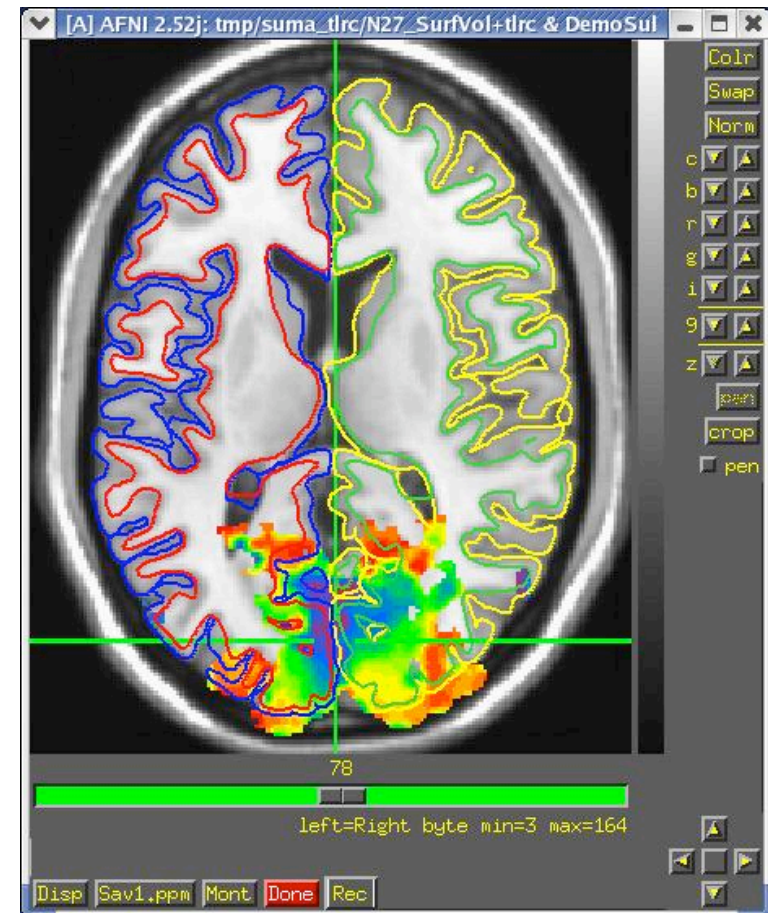
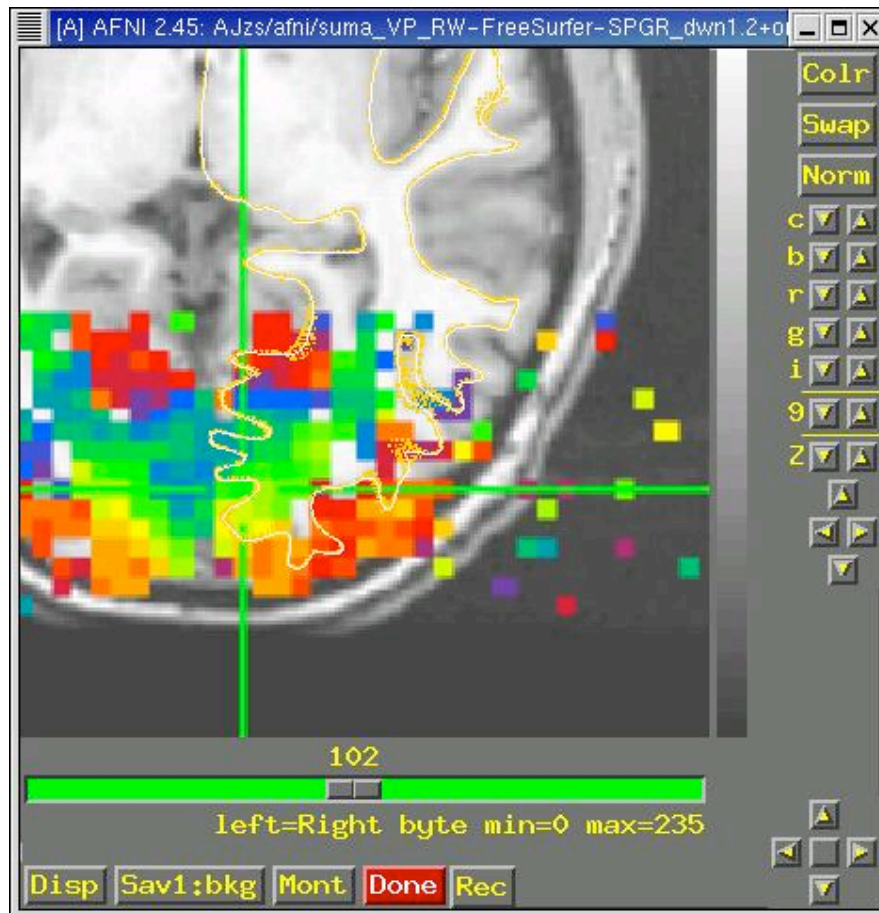
- ➔ You should see a surface overlaid onto DemoSubj_SurfVol_Alnd_Exp+orig
- ➔ Alignment should be proper, otherwise you have a problem.

C: Mapping FMRI Data Onto Surface



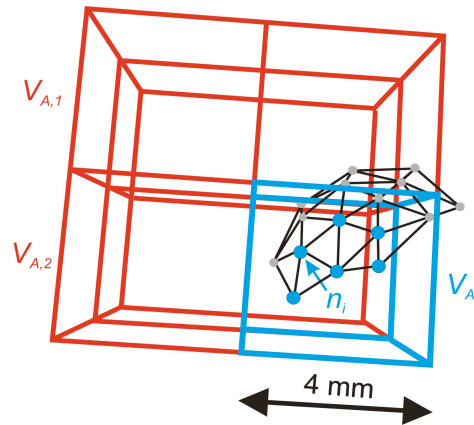
Mapping Options

- Surface/volume Intersection
- Shell/volume Intersection

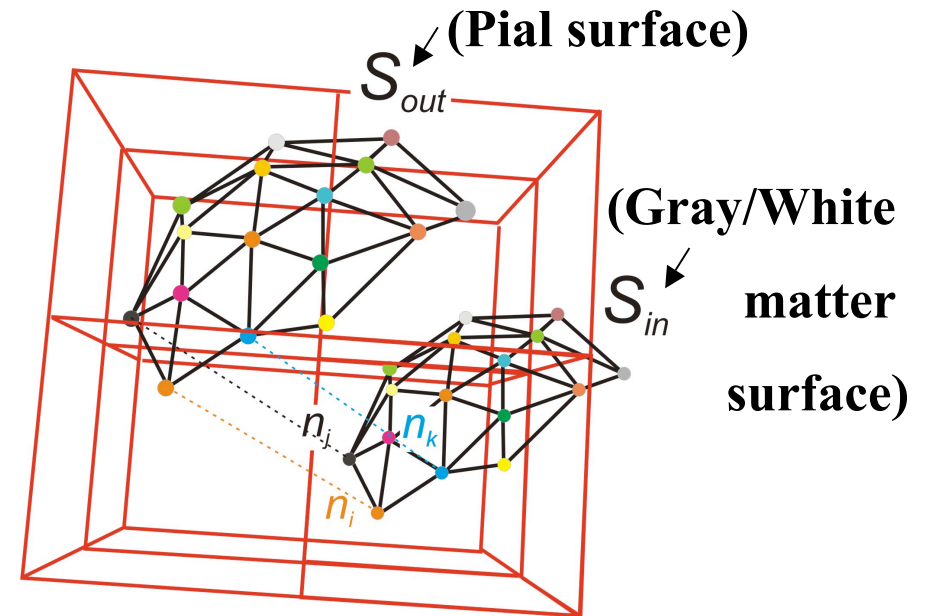


Mapping options

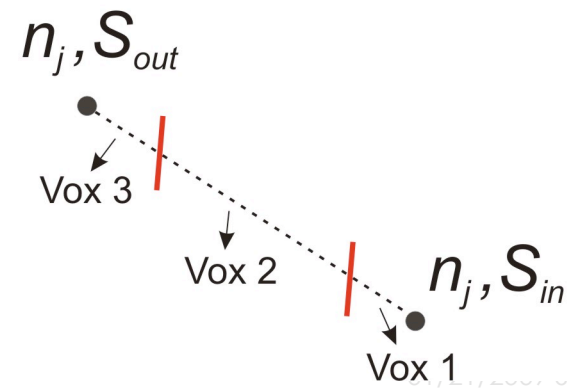
- Surface/volume Intersection
 - ★ One voxel per node



- Shell/volume Intersection
 - Multiple voxels possible per node



Node segment
intersecting 3 voxels



Mapping Habits

- When mapping data from volume to surface domains
 - ★ If mapping assigns multiple voxels to one node
 - ↳ How do you deal with functional data sets?
 - ⇨ Do you average statistics?
 - ⇨ Do you apply a threshold before or after averaging?
 - » What if some of the voxels are active and some are not?
 - ↳ These problems are best avoided by:
 - ⇨ Mapping the time series data onto the surface
 - ⇨ Performing statistical analysis directly in the surface domain using *3dSomething* AFNI programs
- When combining surface data across subjects
 - ★ Same concerns as with volumetric group analysis
- You can also create volumetric data from surface-based data
 - ★ use **3dSurf2Vol**, the reciprocal of **3dVol2Surf**

Interactive Mapping Onto Surface

- Interactive mapping is controlled by AFNI's [Vol2Surf plugin](#)
 - ★ Mapping is done using anatomically correct surfaces sent to AFNI with the overlay data volume.
 - ➔ See plugin's help for details.
 - ★ Only colors (not data values) are sent from AFNI to SUMA.
- Demo (*Continued*): To map data from volume to surface interactively:
 - ★ Switch Overlay to DemoSubj_EccExpavir.DEL
 - ★ Define Overlay with:
 - ➔ Olay: Delay
 - ➔ Thr: Corr. Coef.
 - ➔ Pos. color mapping
 - ➔ #20 color map
 - ➔ See Function
 - ⇒ You should see the function on the surface model in SUMA.
 - ⇒ The colors are applied to all topologically related surfaces
 - ⇒ NOTE: Only AFNI controller A sends function back to SUMA
 - ➔ Change threshold in AFNI and watch change in SUMA

Command-line Mapping

- **3dVol2Surf** maps data directly onto the cortical surface

3dVol2Surf

```
-spec      ../SurfData/SUMA/std.DemoSubj_lh.spec \
-surf_A    lh.smoothwm.asc                      \
-surf_B    lh.pial.asc                          \
-sv        DemoSubj_SurfVol_AInd_Exp+orig       \
-grid_parent DemoSubj_EccExpavir.DEL+orig      \
-map_func  ave                                  \
-f_steps   10                                  \
-f_index   nodes                              \
-out_niml   v2s.DEL.niml.dset
```

Written by Rick Reynolds

(see output of *3dVol2Surf -help* for details.)

- Execute script *tcsh run_3dVol2Surf* to map various types of data onto the surfaces

Options for 3dVol2Surf example

★ Basic Parameters:

- spec: SUMA spec file containing surface(s) to be used in mapping.
- surf_A (-surf_B): Surface(s) to be used in the mapping.
- sv: Surface Volume used to align surface to data
- grid_parent: AFNI volume containing data to be mapped.
- map_func: Method for handling voxels to node mapping
- out_niml: Output data set file in NIML format

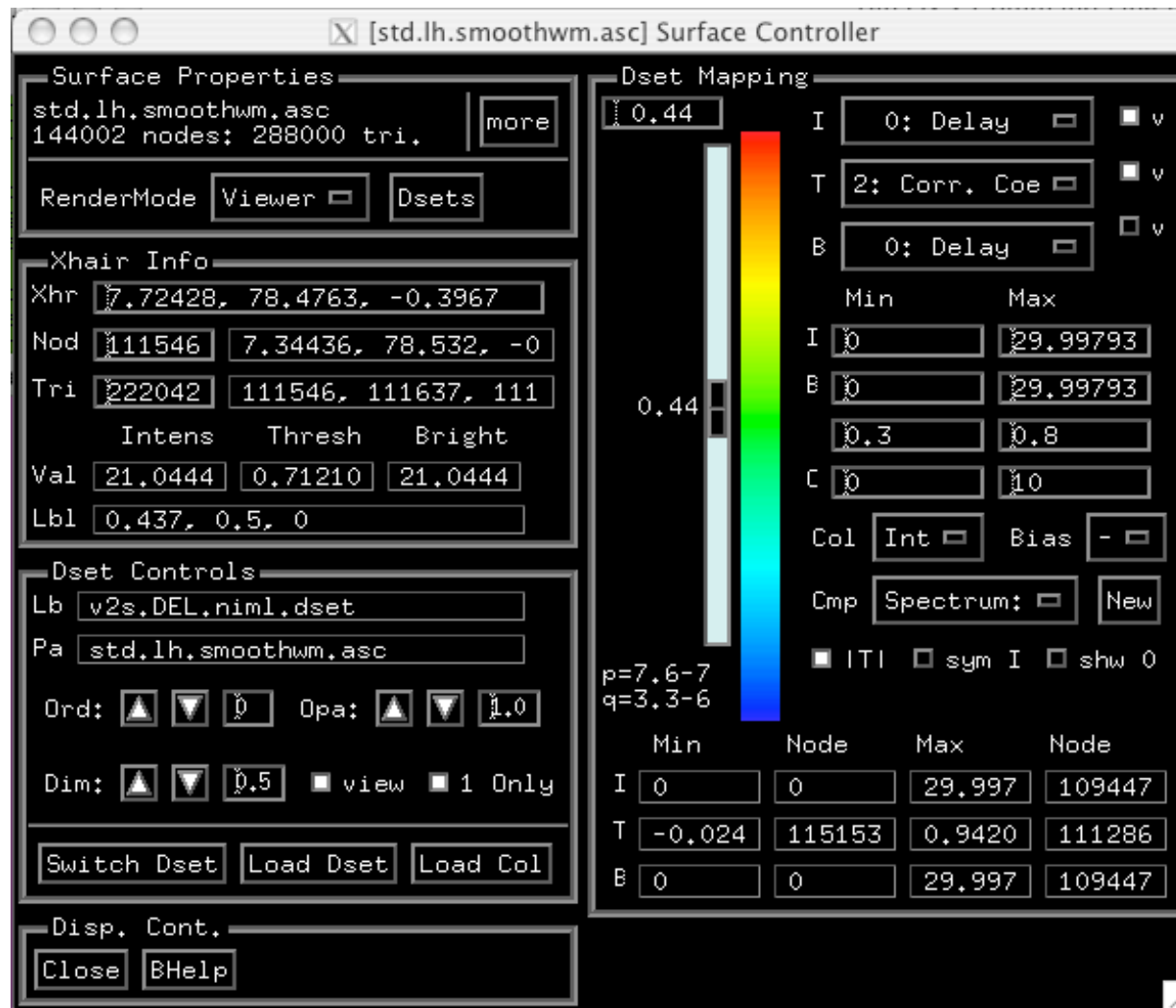
★ Optional parameters (just a few of them):

- cmask: Option for masking data in DataVol on the fly..
- oom_value V0 : Assign V0 to nodes in masked voxels
- oob_value V1: Assign V1 to nodes that are not covered by any voxels
- oom_value and -oob_value are useful for creating 'full' datasets.

Format of surface-based datasets

- The output dataset can be thought of as a table of numbers.
 - Each column represents data from one sub-brick in the volume
 - Each row represents values mapped to a node
 - By default, a surface dataset is 'sparse'; containing only rows for the nodes that had data mapped onto them
 - When combining surface-based datasets, it is safest to work with 'full' datasets which contain a row for each node
 - [ConvertDset](#) can turn 'sparse' datasets into 'full' ones.
- Useful tools for manipulating datasets:
 - [3dinfo](#) (Information as viewed by AFNI's 3d* programs)
 - [SurfDsetInfo](#) (Information as viewed by SUMA)
 - [ConvertDset](#) (A tool to convert between formats)
- Selection qualifiers to dataset names
 - [SEL] to select certain columns (sub-bricks)
 - {SEL} to select certain rows
 - #SEL# to select certain nodes
 - The format of SEL is the same as in AFNI, see section 'INPUT DATASET NAMES' in 3dcalc -help for details.
 - [i] to select the node index column (NIML dsets only)

Colorizing Results Interactively



- Demo (continued):
 - ★ In SUMA, press 'ctrl+s' to open Surface Controller.
 - ➡ Use View→Surface Controller if you were born after 1981.

Graphing Time Series

- Demo (*continued*):

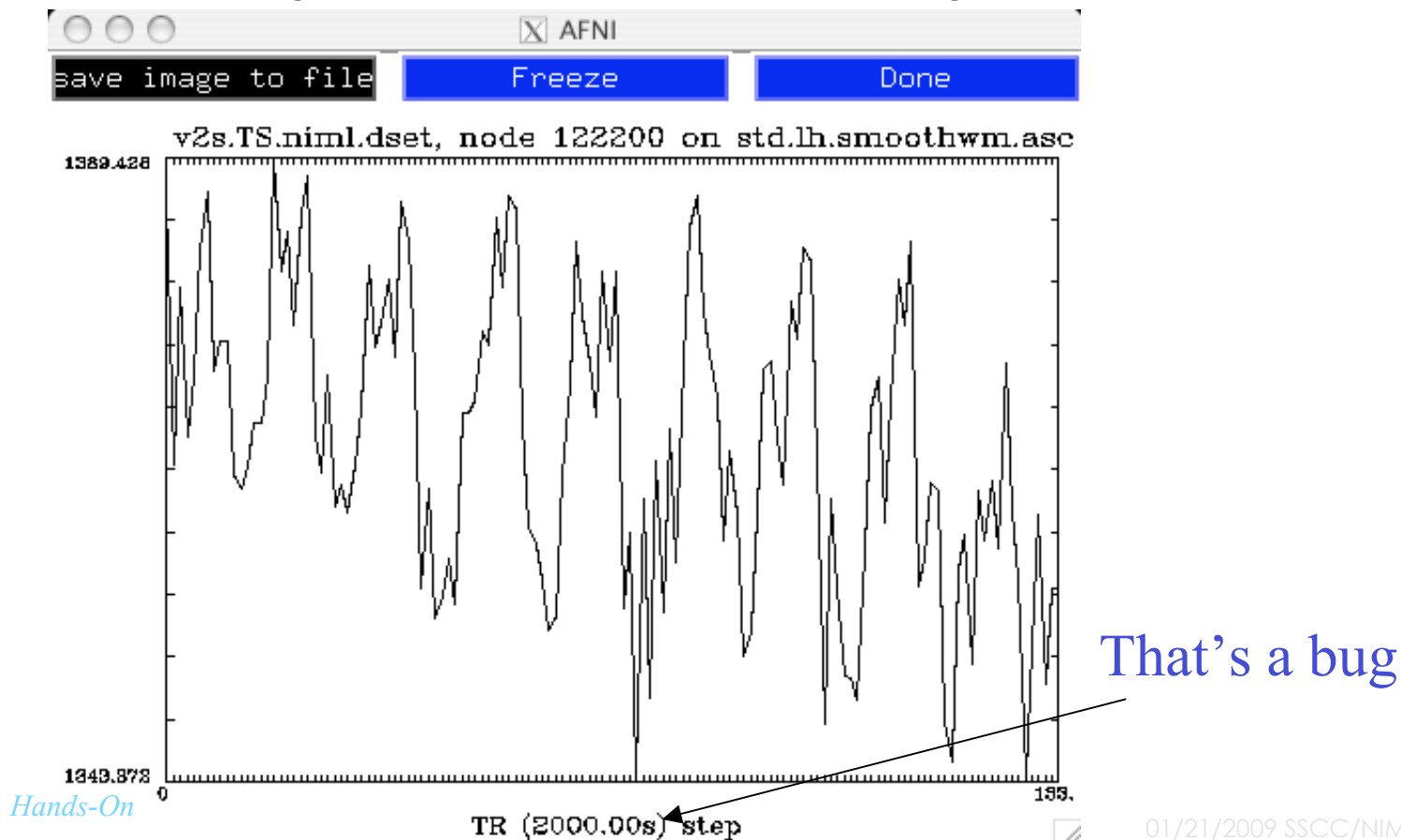
- ★ Press “Load Dset” and read in “v2s.TS.niml.dset”

- In SUMA, press ‘g’ to graph data points at the selected node

- Select other nodes to see data at other nodes

- Press ‘Freeze’ on graph window to preserve current graph

- Clicking on other nodes will start a new graph



Colorizing results interactively

- Demo (*continued*):

- ★ Press “Load Dset” and read in “v2s.DEL.niml.dset”

- ➔ SUMA will colorize the loaded Dset (create a color plane for Dset) and display it on the top of pre-existing color planes.

- ➔ We begin by describing the right side block “Dset Mapping” which is used to colorize a Dset. Many of the options mimic those in AFNI’s “define OverLay” controls.

- ➔ Many features are not mentioned here. See online docs. and BHelp.

- ★ From the Dset Mapping block (right side of interface)

- ➔ Select column Corr. Coef. for Threshold (T)

- ⇒ Press ‘v’ button to apply thresholding

- ⇒ Use scale to set the threshold. Nodes whose cross correlation value does not pass the threshold will not get colored

- ⇒ Note p (uncorrected), and q values (FDR) below the slider

- ⇒ Note:

- ⇒ For simplicity, we mapped a statistical dataset onto the surface. This resulted in statistical parameters being averaged without being normalized

- ⇒ A better approach would be to map the time series, and then perform the statistical computation. See script `./run_3dVol2Surf` for examples.

Dset Mapping block

- Demo (*continued*):
- **Mapping Parameters Table:**
 - ➔ Used for setting the clipping ranges.
 - ➔ Clipping is only done for color mapping. Actual data values do not change.
- ★ Col. *Min*:
 - ➔ Minimum clip value. Clips values (v) in the Dset less than Minimum (\min): if $v < \min$ then $v = \min$
- ★ Col. *Max*:
 - ➔ Maximum clip value. Clips values (v) in the Dset larger than Maximum (\max): if $v > \max$ then $v = \max$
- ★ Row *I*
 - ➔ Intensity clipping range. Values in the intensity data that are less than Min are colored by the first (bottom) color of the colormap. Values larger than Max are mapped to the top color.
- ★ Left click locks ranges from automatic resetting.
- ★ Right click resets values to full range in data.

Dset Mapping block

- Demo (*continued*):
 - **Col:**
 - Switch between color mapping modes.
 - *Int*: Interpolate linearly between colors in colormap
 - *NN* : Use the nearest color in the colormap.
 - *Dir*: Use intensity values as indices into the colormap. In *Dir* mode, the intensity clipping range is of no use.
 - **Cmp:**
 - Switch between available color maps. If the number of colormaps is too large for the menu button, right click over the 'Cmp' label and a chooser with a slider bar will appear.
 - Alternately, as with many of SUMA's menus, detach the menu by selecting the dashed line at the top of the menu list. Once detached, the menu window can be resized so you can access all elements in very long lists.
 - More help is available via ctrl+h while mouse is over the colormap
- Bias:

Dset Mapping block

- Demo (*continued*):
- **The Colormap:**
 - The colormap is actually a surface in disguise and shares some of the functions of SUMA's viewers:
 - Keyboard Controls:
 - r: record image of colormap.
 - Ctrl+h: this help message
 - Z: Zoom in.

Maximum zoom in shows 2 colors in the map
 - z: Zoom out.

Minimum zoom in shows all colors in the map
 - Up/Down arrows: move colormap up/down.
 - Home: Reset zoom and translation parameters
 - Mouse Controls:
 - None yet, some maybe coming.

Dset Mapping block

- Demo (*continued*):
 - **|T|:**
 - ★ Toggle Absolute thresholding.
 - ➔ OFF: Mask node color for nodes that have:
 $T(n) < T_{scale}$
 - ➔ ON: Mask node color for nodes that have:
 $|T(n)| < T_{scale}$
- where: T_{scale} is the value set by the threshold scale.
 $T(n)$ is the value in the selected threshold column (T).
- **sym I:**
 - ★ Toggle Intensity range symmetry about 0.
 - ➔ ON : Intensity clipping range is forced to go from -val to val . This allows you to mimic AFNI's ranging mode.
 - ➔ OFF: Intensity clipping range can be set to your liking.
 - **shw 0:**
 - ★ Toggle color masking of nodes with intensity = 0
 - ➔ ON : 0 intensities are mapped to the colormap as any other values.
 - ➔ OFF: 0 intensities are masked, a la AFNI

Dset Mapping block

- Demo (*continued*):
- **Data Range Table:**
 - ★ Full range of values in Dset.
 - ➔ Right click in “Node” columns to have cross hair jump to that node’s location.

Xhair Info block

- Demo (*continued*):
- **Xhr:**
 - ★ Crosshair coordinates on this controller's surface. Entering new coordinates makes the crosshair jump to that location (like 'ctrl+j').
 - ➔ Use 'alt+l' to center the cross hair in your viewer.
- **Node:**
 - ★ Node index of node in focus on this controller's surface. Nodes in focus are highlighted by the blue sphere in the crosshair. Entering a new node's index will put that node in focus and send the crosshair to its location (like 'j').
- **Node Values Table:**
 - ★ Data Values at node in focus
 - ➔ Col. *Intens*: Intensity (I) value
 - ➔ Col. *Thresh*: Threshold (T) value
 - ➔ Col. *Bright*: Brightness modulation (B) value
 - ➔ Row. *Val*: Data Values at node in focus
- **Node Label Table:**
 - ★ Row. *Lbl*:
 - ➔ Color from the selected Dset at the node in focus. For the moment, only color is displayed. The plan is to display labels of various sorts here.

Dset Controls block

- Demo (*Continued*):
- **Dset Info Table:**
 - ★ Row. *Lbl*: Label of Dset.
 - ★ Row. *Par*: Parent surface of Dset.
- **Ord:**
 - ★ Order of Dset's colorplane. Dset with highest number is on top of the stack. Separate stacks exists for foreground (fg:) and background planes (bg:).
- **Opa:**
 - ★ Opacity of Dset's colorplane. Opaque planes have an opacity of 1, transparent planes have an opacity of 0. Opacities are used when mixing planes within the same stack foreground (fg:) or background(bg:).
 - ★ Opacity values are not applied to the first plane in a group. Consequently, if you have just one plane to work with, opacity value is meaningless.
 - ★ Color mixing can be done in two ways, use F7 to toggle between mixing modes.

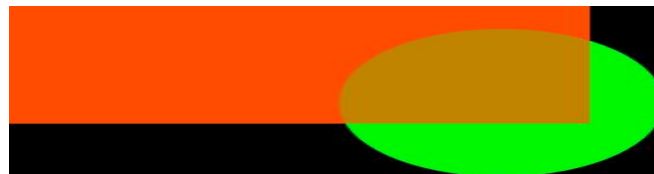
Dset Controls block

- Demo (*Continued*):
- **Dim:**
 - ★ Dimming factor to apply to colormap before mapping the intensity (I) data. The colormap, if displayed on the right, is not visibly affected by Dim but the colors mapped onto the surface are.
 - ★ For RGB Dsets (.col files), Dim is applied to the RGB colors directly.
- **view:**
 - ★ View (ON)/Hide Dset node colors.
- **1 Only:**
 - ★ If ON, view only the selected Dset's colors. No mixing of colors in the foreground stack is done.
 - ★ If OFF, mix the color planes in the foreground stack.
 - ★ This option makes it easy to view one Dset's colors at a time without having to worry about color mixing, opacity, and stacking order.
 - ★ Needless to say, options such as 'Ord:' and 'Opa:' in this panel are of little use when this button is ON.

Color overlay planes

- Colorized Dsets are organized into layered color planes
 - ★ 2 commonly used planes are:
 - ↳ Surface Convexity (usually in gray scale)
 - ↳ AFNI Function (usually in color)
 - ★ Planes are assigned to two groups
 - ↳ Background planes (like Convexity)
 - ↳ Foreground planes (like AFNI Function)
 - ★ Many other planes can be added to either group.
- Color planes of the same group are mixed together:
 - ★ Planes are stacked based on their order and opacity.
 - ★ Opacity of 1st plane does not affect color mixing.
 - ★ There are 2 modes for mixing colors. See option F7 in SUMA.

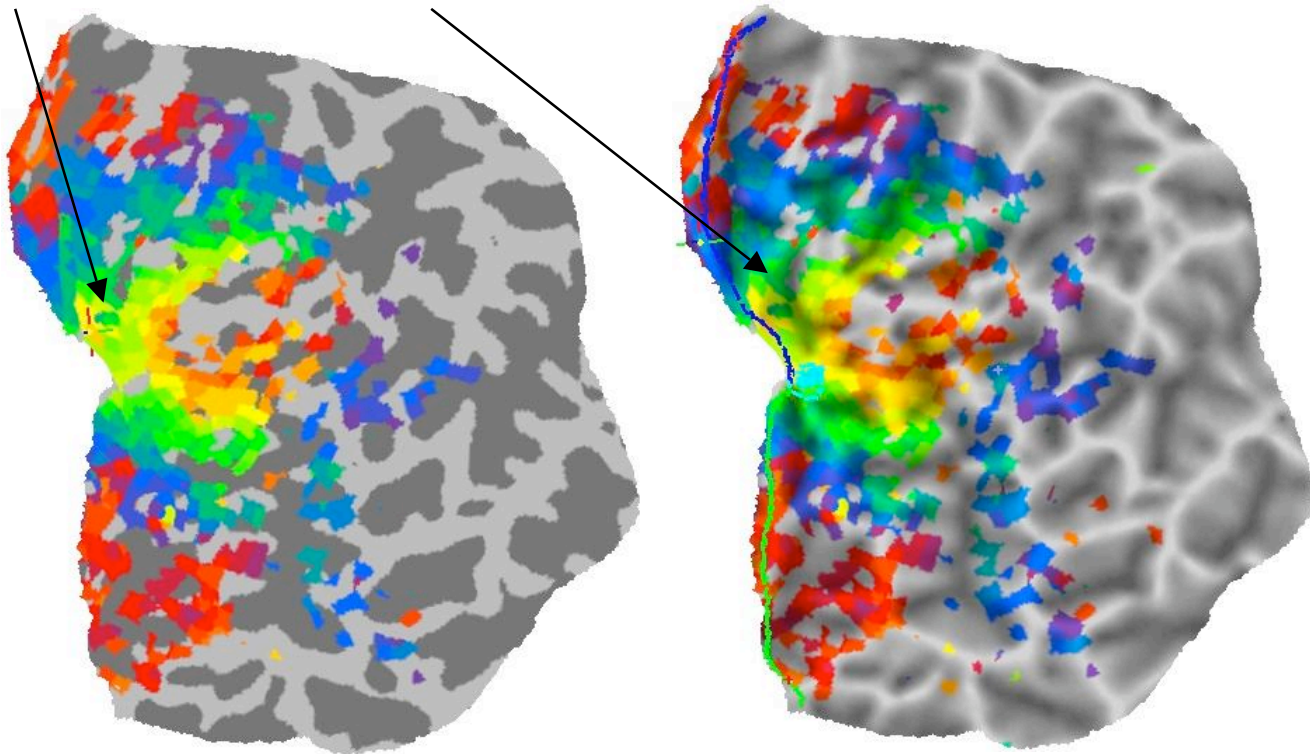
Plane 1: 80% opacity



Plane 2: 30% opacity

Color overlay planes

- Node colors displayed on surface are obtained by:
 - ★ 1st: mixing background planes
 - ★ 2nd: mixing foreground planes
 - ★ 3rd: layering mixed foreground atop mixed background plane
 - ➔ When foreground colors overlap background colors, they either mask or get attenuated by the background's brightness.



Layering mixed fore- & background plane

- Demo (*continued*):
 - ★ View the surface in its inflated state with AFNI function
 - ★ Turn foreground plane(s) off by pressing 'f' once
 - ➔ Now all you see is the background plane(s)
 - ★ Turn background planes off by pressing 'b' once
 - ➔ Now all you see is the No Color color on all nodes
 - ★ Turn foreground plane(s) back on with 'f'
 - ➔ Now you have foreground without background
 - ★ Turn background plane(s) back on with 'b'
 - ➔ Now you have foreground atop background
 - ➔ Notice how you can still see the background underneath the foreground—this is due to the background brightness attenuation of the foreground colors.
 - ★ Toggle background intensity attenuation off and on with 'a' and see the effect on the resultant maps.

Playing with color plane opacity

- Demo (*continue from inflated view with function*)
 - ★ 'ctrl+s' or View → Surface Controller to open the surface controller
 - ↳ Turn OFF '1 Only'
 - ★ Load in color plane lh.1D.col with 'Load Col' or 'c'
 - ↳ This is an RGB Dset, color mapping controls are hidden
 - ↳ Plane is placed atop of the foreground group
 - ↳ Its opacity is 1 so it will obscure the functional data
 - ↳ Background attenuation is not affected by plane's opacity.
 - ↳ try turning it on and off again with 'a'
 - ↳ Now lower the opacity of lh.1D.col and watch the colors from the planes below start to show through

Playing with color plane order

- Demo *(continue from inflated view with function)*
 - ★ You could put lh.1D.col below the function
 - Switch Color Plane to get a list of available planes
 - Prefixes fg: and bg: denote plane's group membership
 - Select lh.1D.col and lower its order
 - Select FuncAfni_0 and play with its opacity
 - Note: You can't make a plane change its group membership, yet.
 - ★ You can't delete a loaded color plane yet, but you can hide it.
 - ★ Turn '1 Only' ON if you just want to see the selected plane.
- Test
 - ★ Find a way to flip between the mapping from AFNI and the mapping done with 3dVol2Surf before.
 - Appreciate the differences between the two mappings.

Drawing surface-based ROIs

- Demo
 - ★ 'Ctrl+d' or Tools → Draw ROI to open ROI drawing tool
 - ↳ When in Draw ROI mode (cursor turns into target circles)
 - ⇨ The pick (usually third) mouse button is used for drawing
 - ⇨ Picking is done by combining shift key and pick button
 - ↳ If pen mode is selected (cursor turns into a pen)
 - ⇨ Drawing and picking (with shift) are done with the 1st mouse button and rotations are done with the third button
 - ★ When you draw for the first time, a new “drawn ROI” is created.
 - ↳ Note the Parent: field in the ROI frame is now filled

Drawing surface-based ROIs

- Demo
 - ★ Move the mouse to a new location and click again to draw a line from the location of the first click to a new one.
 - ★ Or click while dragging the mouse to create a smoother line.
 - ★ Both drawing methods might fail if you are drawing over very rough terrain.
 - ➔ When that happens, you continue from where the drawing stopped.
 - ➔ Drawing in 3D looks easy but it isn't.
 - ★ Use Undo and Redo when you get your drawings messed up.
 - ★ To close a loop you can press 'Join' button or double click

Drawing surface-based ROIs

- Demo (*Continued*)
 - ★ To fill a closed loop, click inside the loop.
 - ➔ Note: If you have patterns that make a figure 8 jealous you might have to do multiple fills.
 - ★ Press Finish when you are done drawing and have set the drawing label and value to your liking.
 - ★ Now you can draw another ROI, and another and another.
 - ★ Drawing can be started/continued on any of the related surfaces
 - ➔ When you hit 'join', the loop is closed using the surface where the ROI was created
 - ★ Switch ROI to switch between ROIs and delete/modify them.
 - ★ Load to load ROIs from file
 - ★ Save to save ROIs to file
 - ➔ 1D/Niml Format of ROI file
 - ➔ This/All which ROI to save into the file

Surface ROI → Volume ROI

- ROIs files should be transformed to dataset files (Dset).
 - ★ A dataset file has an equal number of values for each node
 - ★ The way we drew the two ROIs, certain nodes belong to more than one ROI (i.e. they have more than one value)
- The program **ROI2dataset** is used to change ROIs to datasets
- Execute: *tcsh run_ROI2dataset*

```
ROI2dataset  -prefix OccROIs.1D.dset      \  
              -of      1D                  \  
              -input  OccROIs.niml.roi
```

- Note the warning:

Warning SUMA_ROIv2dataset:

155/2536 nodes had duplicate entries.

(ie same node part of more than 1 ROI)

Duplicate entries were eliminated.

- ★ At the moment, duplicate entries are ignored, 1st come, 1st adopted. More options could be added if you find necessary.

Surface ROI → Volume ROI

- OccROIs.1D.dset can be transformed into a volume ROI with 3dSurf2Vol
- Execute: *source run_3dSurf2Vol*

```
3dSurf2Vol -spec../SurfData/SUMA/DemoSubj_lh.spec \
          -surf_A      lh.smoothwm.asc \
          -surf_B      lh.pial.asc \
          -sv          DemoSubj_SurfVol_AInd_Exp+orig \
          -grid_parent DemoSubj_SurfVol_AInd_Exp+orig \
          -map_func     max \
          -f_steps      10 \
          -f_p1_mm      -0.5 -f_pn_fr 0.5 by Rick Reynolds \
          -sdata_1D     OccROIs.1D.dset \
          -prefix       OccROIs
```

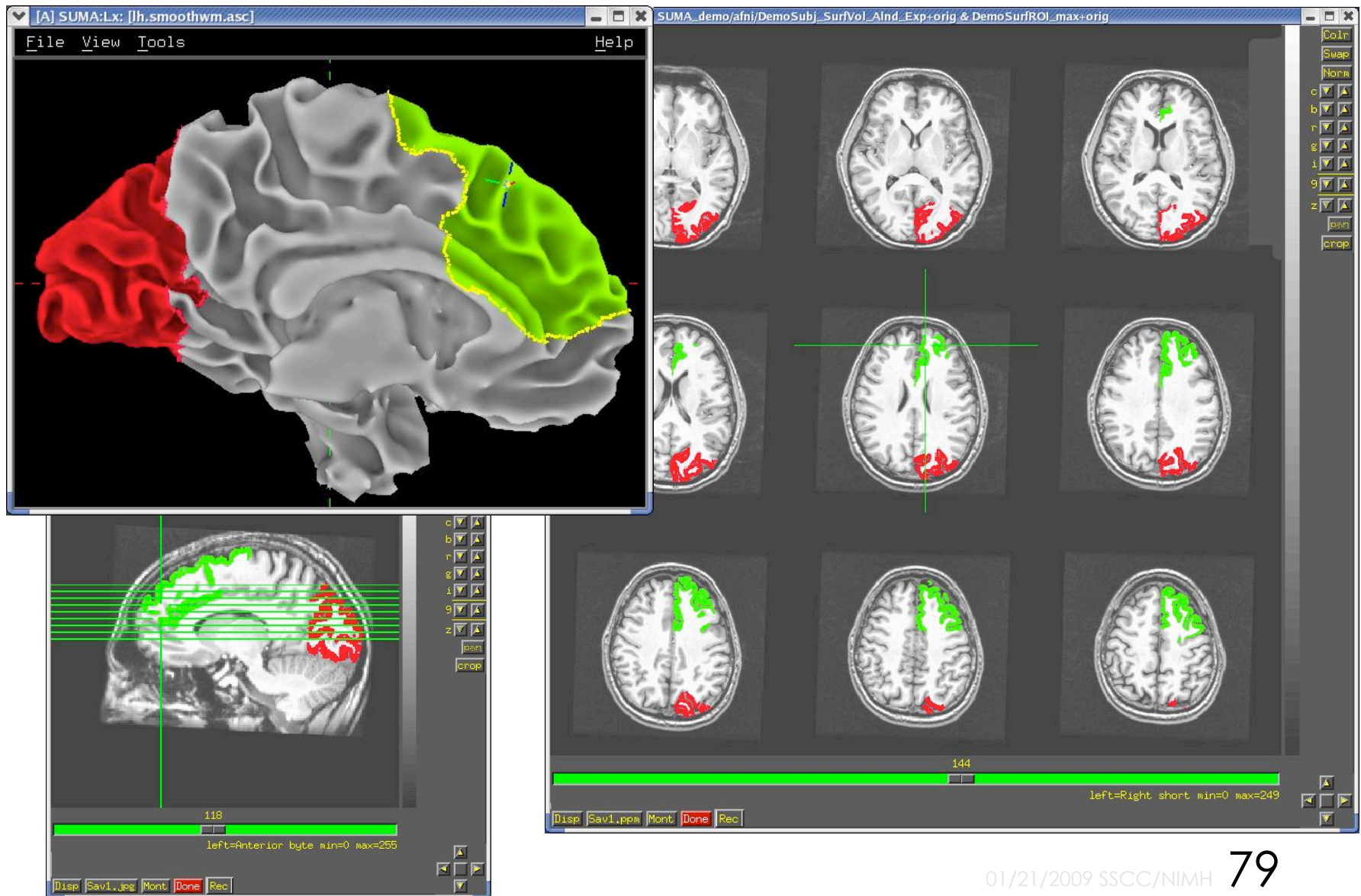
-grid_parent specifies the output volume's geometric properties

-f_p1_mm, -f_pn_fr: Specify extensions on node pair segment, either in mm or fractions of segment length (both used simultaneously here for illustration)

-sdata_1D: Specify surface data file

Surface ROI → Volume ROI

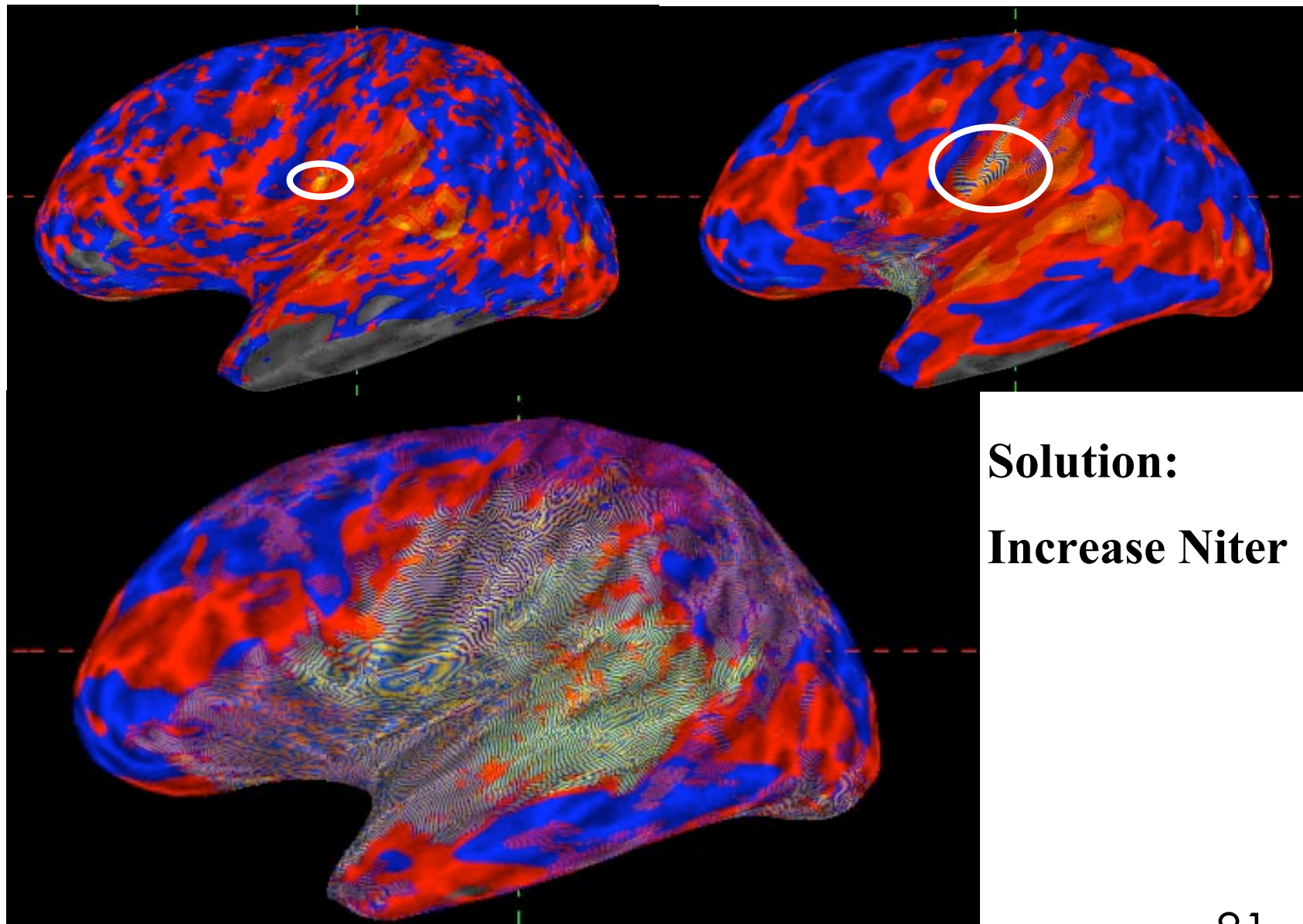
check your own OccROIs+orig in AFNI



Smoothing data on surface models

- Data smoothing (along the surface) is done with **SurfSmooth**
 - ★ The difficulty in smoothing lies in calculating geodesic distances between node pairs.
 - ★ Smoothing can be performed by solving the diffusion equation on the surface. However effect of kernel is is somewhat poorly estimated. Number of iterations needed is a wild guess with numerical accuracy issues. Better smooth until a certain smoothness is reached.
 - ★ SurfSmooth can estimate smoothness
 - ★ until desired smoothness is reached
 - ★ or until additional smoothness is achieved
- Demo (*close SUMA and afni*)
 - ★ Execute: *tcsh run_SurfSmooth_data*
 - The script will use the output of *3dVol2Surf* in *run_3dVol2Surf* to map timeseries data onto the surfaces
 - SUMA is then launched and the script waits for you to setup the recorder ON
 - Hit enter twice (in the shell) to launch the smoothing program
 - Watch the smoothing progress with each iteration. You can play the whole sequence at the end.

Look At Your Data!



Solution:

Increase Niter

Smoothing geometry of surface

- Surface geometry smoothing can also be done with **SurfSmooth**
 - ★ The difficulty in geometry smoothing is in preventing the surface from shrinking in size.
 - ★ **SurfSmooth** uses the smoothing algorithm by (Taubin G. 2000)
- Demo (*close SUMA and AFNI*)
 - ★ Execute: `tcsh run_SurfSmooth_geom`
 - ➔ The script will add noise to the smoothwm surface, then filter it
 - ➔ SUMA is then launched and the script waits for you to setup
 - ❏ Turn off background colors ('b')
 - ❏ Switch to noisy surface ('.')
 - ❏ Turn recorder ON
 - ➔ Hit enter twice (in the shell) to launch the smoothing program
 - ➔ Watch the smoothing progress with each iteration. You can play the whole sequence at the end.

Talairach data display, for panache

- Without creating individualized surfaces, you can display data on a Talairach-ed surface model.
 - ★ The surface models were created with FreeSurfer from the N27 data set. (Holmes, CJ et al. JCAT 1998)
 - ★ Anatomical volume and surfaces were Talairach-ed using AFNI and SUMA, respectively. (Surfaces created by Brenna Argall)
- Demo: (close previous SUMA and AFNI sessions)
 - ★ *cd suma_demo/afni_tlrc*
 - ➔ You should find Anat+tlrc and a functional data set in Talairach space
 - ★ *afni -niml &*
 - ➔ setup function as shown in the earlier slides
 - suma*
 - spec ../SurfData_tlrc/SUMA/N27_lh_tlrc.spec *
 - sv Anat+tlrc*
 - ➔ start communication with AFNI
 - ➔ map function onto surface
 - ➔ appreciate the limitation of this approach
 - ➔ ignore the limitation and make cool pictures

Auxiliary programs

- To see a list of all programs in SUMA:
 - ★ `suma -progs`
- `3dSurf2Vol` (by R. Reynolds)
 - ★ Maps data in surface domain volumetric domain
- `3dVol2Surf` (by R. Reynolds)
 - ★ Maps data in volume domain to surface domain
- `CompareSurfaces` (by S. Japee)
 - ★ calculates the distance along the normal from one surface to the next
- `ConvertSurface`
 - ★ Converts surfaces between the different formats that SUMA can read:
 - ➔ FreeSurfer
 - ➔ SureFit
 - ➔ Simple ASCII matrix format
 - ➔ PLY format
 - ★ Converts surface coordinates to Talairach space
- `ConvexHull`
 - ★ Finds the convex hull of a volume or a set of points

Auxiliary programs

- **Createlcosahedron** (by B. Argall)
 - ★ Creates Icosahedral meshes of varying node counts
- **inspec**
 - ★ Outputs information about a spec file
- **IsoSurface**
 - ★ Creates an isosurface from a volumetric data set
- **MakeColorMap**
 - ★ Creates colormaps for use with AFNI
- **Maplcosahedron** (by B. Argall)
 - ★ transforms the topology of surface models to standard models without distorting the geometry (tested with FreeSurfer's spherical mapping).
 - ★ use to check for errors in topology of spherical surfaces
- **quickspec**
 - ★ Creates a spec file for one or a set of surfaces (quick and dirty)
- **ROI2dataset**
 - ★ Transforms ROI files into surface data sets

Auxiliary programs

- **ScaleToMap**
 - ★ Transforms a set of node values to node colors based on chosen color map
- **SurfaceMetrics**
 - ★ Outputs other information about the surface such as the edge list, curvature, triangle areas (mostly for debugging use)
- **SurfClust**
 - ★ Clustering on the surface
- **SurfMeasures** (by R. Reynolds)
 - ★ Outputs node-based surface measures such as areas, volumes, thickness, surface normals, etc. etc.
- **SurfSmooth**
 - ★ a program for filtering surface data and/or surface geometry
- **SurfPatch**
 - ★ Creates surface patches from a set of nodes.
- **SurfQual**
 - ★ Locates topological errors in spherical surfaces.

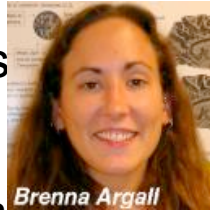
Getting Helped

- suma –help for SUMA's command line usage
- 'ctrl+h' opens a window with help for:
 - ★ SUMA's usage when cursor is in viewer
 - ★ SUMA's colormap usage when cursor is over colormap
- BHelp provides help for most buttons in the GUI interface
- AFNI's web site:
 - ★ <http://afni.nimh.nih.gov/>
 - ★ <http://afni.nimh.nih.gov/afni/community/board/list.php?f=1>
- AFNI's Message Board
- E-mail to saadz@mail.nih.gov
- If you can't get help here, please get help somewhere.

Thanks to:

Developers:

Brenna Argall
Shruti Japee
Rick Reynolds



Critics:

Mike Beauchamp
Pat Bellgowan



Data and Documentation
Peggy Christidis

Getting Help Somewhere
Richard Doucette



Testing: Samia Saad

SysAdmin:

Alex Clark



MRI Pulse Sequence
Segmentation
Hauke Heekeren
Sean Marrett

